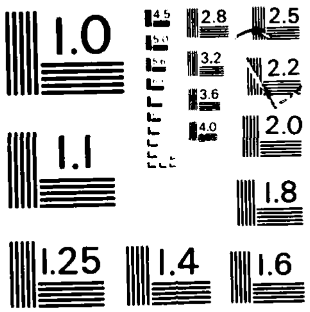
AD-A132 063 PROCEDURES FOR THE TRANSFER OF DATA FILES BETWEEN THE BRL CYBER COMPUTER..(U) ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND ABERDEEN PROVI.. C L HENRY AUG 83 1/1

UNCLASSIFIED ARBRL-MR-03298 F/G 9/2 NL



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS - 1963 - A

(12)

AD F300302

MEMORANDUM REPORT ARBRL-MR-03298

# PROCEDURES FOR THE TRANSFER OF DATA FILES BETWEEN THE BRL CYBER COMPUTER SYSTEM AND THE IBD HEWLETT-PACKARD MINICOMPUTER NETWORK

Caledonia L. Henry

August 1983

US ARMY ARMAMENT RESEARCH AND DEVELOPMENT COMMAND
BALLISTIC RESEARCH LABORATORY
ABERDEEN PROVING GROUND, MARYLAND

Approved for public release; distribution unlimited.

Destroy this report when it is no longer needed.
Do not return it to the originator.

Additional copies of this report may be obtained from the National Technical Information Service, U. S. Department of Commerce, Springfield, Virginia 22161.

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

*The use of trade names or manufacturers' names in this report does not constitute indorsement of any commercial product.*

| | | |
|---|---|---|
| **REPORT DOCUMENTATION PAGE** | | READ INSTRUCTIONS<br>BEFORE COMPLETING FORM |
| 1. REPORT NUMBER<br>MEMORANDUM REPORT ARBRL-MR-03298 | 2. GOVT ACCESSION NO.<br>AD-A132 003 | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle)<br>PROCEDURES FOR THE TRANSFER OF DATA FILES BETWEEN THE BRL CYBER COMPUTER SYSTEM AND THE IBD HEWLETT-PACKARD MINICOMPUTER NETWORK | | 5. TYPE OF REPORT & PERIOD COVERED<br>Final Report |
| | | 6. PERFORMING ORG. REPORT NUMBER |
| 7. AUTHOR(s)<br>CALEDONIA L. HENRY | | 8. CONTRACT OR GRANT NUMBER(s) |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS<br>US Army Ballistic Research Laboratory<br>ATTN: DRDAR-BLI<br>Aberdeen Proving Ground, MD 21005 | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS<br>1L162618AH80 |
| 11. CONTROLLING OFFICE NAME AND ADDRESS<br>US Army Armament Research & Development Command<br>US Army Ballistic Research Laboratory (DRDAR-BLA-S)<br>Aberdeen Proving Ground, MD 21005 | | 12. REPORT DATE<br>August 1983 |
| | | 13. NUMBER OF PAGES<br>72 |
| 14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office) | | 15. SECURITY CLASS. (of this report)<br>UNCLASSIFIED |
| | | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

minicomputer
microcomputer
main frame
data transfer
BASIC
FORTRAN IV

20. ABSTRACT (Continue on reverse side if necessary and identify by block number) jmk

The need for data transfer between the Ballistic Research Laboratory CYBER computer system and the Interior Ballistics Division Hewlett-Packard minicomputer system exists. This is a method for on-line data transfer with conversion programs at both ends of the transfer.

## TABLE OF CONTENTS

## I. INTRODUCTION

Frequently, files stored on one computer must be transferred to another computer. This need has become more prevalent in the last few years with the advent of satellite minicomputer networks within the Ballistic Research Laboratory (BRL).

In particular, the Interior Ballistics Division (IBD) of the BRL has established a minicomputer network with a HEWLETT-PACKARD (HP) 1000-F minicomputer at the hub. Experimental data recorded on analog tape are converted to digital form and stored on digital tape and/or disk for further processing. This conversion is accomplished using an HP analog-to-digital converter. These data may be transferred from the HP1000 to one of the several HP9845 microcomputers within the IBD for intermediate or final analysis.

•If analysis is desired on a larger machine, the data may be transferred to the BRL's Control Data Corp (CDC) computer mainframe system. This system is composed of a CDC CYBER 173 (MFA), a CDC CYBER 825 (MFB) and a CDC CYBER 7600 (MFZ). MFA is the common link to MFB and MFZ, and, therefore, is the transfer point for the CDC system. •

After the data analysis is finished on the CDC system, output data files are sent to the minicomputer or microcomputer site for such things as interactive computer graphics processing or additional analysis using existing software. So, the transfer from MFA to the HP9845 is also required.

The method herein discussed has proven to be a convenient means of transferring files (primarily data files) between an HP9845 and MFA.

## II. PREPARATION OF DATA FILES CREATED ON HP9845

The great majority of the data files now created on an HP9845 is stored in binary format. In order to be transferred to MFA, the data must be converted into ASCII character strings. There are currently two Enhanced BASIC programs written for an HP9845 for the purpose of converting these data files into 80-character (column) card images for transfer to MFA. These 80-character card images are called records.

The first conversion program (CNVTB1:C) creates a data file which conforms to a proposed uniform format for experimental data.[1] Both data file formats are designed to be used in FORTRAN digital computer programs that have been written for the conversion into engineering units of and for the further analysis of experimental and simulation data. A listing of this program is found in Appendix A.

The second of these programs (CNVTB4:C) creates a file format which emulates the data files created using the FORTRAN programs ADTAPE3[2] and

[1] *Franz R. Lynn, "A Uniform Format For Experimental Data," BRL Memorandum Report ARBRL-MR-03151, December 1981 (AD 109964).*

[2] *E.M. Wineholt, C.L. Henry, K.L. Zimmerman, "A Procedure For the Semiautomatic Reduction of Experimental Data," BRL Memorandum Report ARBRL-MR-03259, April 1983 (AD A127321).*

ADTAPE4. Program ADTAPE3 creates a data file accepted by FORTRAN program ADTAPE4 which converts data into engineering units. If the data are already in engineering units, they may be used in other data analysis programs which use output from program ADTAPE4. In most cases, the data transferred from an HP9845 are already in engineering units. Appendix B contains the listing of program CNVTB4.

## A. Data File Format Created Using BASIC Program CNVTB1

An example of a data file which conforms to a uniform format for experimental data is found in Figure 1. This data format is accepted by a modified version (BALDPRESS) of the FORTRAN program BROAD,[3] which converts data to engineering units and integrates and/or plots. This data input sequence is also accepted by other data analysis programs.

The program control and channel information data are found in records one through five. Table 1 provides a description of the variables contained on these records. There is a zero placed in column 80 of records one through five so that all records will have exactly 80 columns.

The format for the data is included in record 2. This format must specify exactly 80 columns of data per card image. If the last record of the data does not contain 80 columns, then the balance of the record will be filled with zeroes so that each record will be 80 columns long.

Figure 1 shows records 1 through 10 and the last five data records. Following the program control and channel information are the first five data records. These are separated from the last five data records by three blank lines.

## B. Data File Format 2 Created Using BASIC Program CNVTB4

Program CNVTB4 is designed to convert one channel or multiple channels of data to be stored on one file. The hardcopy output from this program lists each channel individually and also the summary of the entire file. The individual channels are written to an array and the entire file is stored at one time. This eliminates the necessity of using extra disk space for intermediate storage.

Figure 2 gives an example of a file created to be input for program ADTAPE4 and other data analysis programs. There are three lines of file information preceding the data lines. The variables written in records 1 through 3 are described in Tables 2 and 3. There is a one placed in column 80 of the first three records to force each record to equal 80 columns. The data records are written in eight ten-column fields of fixed point data. If the number of data points is not exactly divisible by eight, the balance of the last data record is filled with zeroes in order to make it 80 columns long.

[3] *C.L. Henry, R.L. Martz, E.M. Wineholt, "An Improved Procedure For The Reduction of Interior Ballistic Data Recorded on Analog Tape," BRL-MR-2374, April 1974 (AD 919923L).*

```
    237MM ACCURACY PROGRAM         10DEC82   1430      31       1              0
LERF          1    0      2048          .0005             0.    8  (8F10.1)    0
UP-DOWN FRONT       DISPLACEMENT IN     MUZZLE SCHMIDT          0    0         0
             0.             0.             1.              0.            0.    0
             1.             0.                                                 0
     125.0     124.0     124.0     125.0     125.0     125.0     125.0     125.0
     124.0     125.0     125.0     125.0     125.0     124.0     125.0     126.0
     125.0     124.0     125.0     125.0     125.0     125.0     125.0     125.0
     125.0     124.0     124.0     125.0     125.0     124.0     126.0     125.0
     125.0     125.0     125.0     126.0     125.0     125.0     125.0     125.0
      •
      •
      •
     115.0     115.0     114.0     114.0     114.0     114.0     114.0     113.0
     113.0     112.0     113.0     113.0     114.0     112.0     112.0     112.0
     113.0     113.0     113.0     113.0     113.0     114.0     114.0     114.0
     115.0     115.0     115.0     115.0     115.0     114.0     116.0     115.0
     115.0     115.0     115.0     117.0     117.0     118.0     125.0     125.0
```

**Figure 1. An Example of a Uniform Format for Experimental Data**

**TABLE 1. DEFINITION OF VARIABLES ON DATA INFORMATION LINES FOR THE UNIFORM FORMAT FOR EXPERIMENTAL DATA**

| Record Number | Variable Name | Column Numbers | Format | Description |
|---|---|---|---|---|
| 1 | IFV | 1-5 | 15 | File format version number, currently 2, but may be changed if the format should be changed. |
| | TITL | 6-35 | A30 | Title of the firing project. |
| | DATE | 36-45 | A10 | Date of the firing. |
| | TIME | 46-55 | A10 | Time of the firing. |
| | IROUND | 56-60 | 15 | Identification number on the analog tape. |
| | IADC | 61-65 | 15 | The number of recording devices used in this firing. |
| 2 | IA | 1-10 | A10 | The name of the recording device used for this particular phenomenon. |
| | NCH | 11-15 | 15 | The number of channels in this group. |
| | ITZ | 16-20 | 15 | Number of calibration data points. |
| | NBD | 21-30 | I10 | Number of phenomenon data points. |
| | *DELT* | *31-45* | *E15** | *Time between consecutive data points.* |
| | OFFTIM | 46-60 | E15* | Offset time from time zero of the data channel, usually the time of the first data point of this channel. |
| | NUMBER | 61-65 | 15 | Number of data points per card image. |
| | IRDFMT | 66-75 | A10 | Format for reading calibration and phenomenon data points. |
| 3 | NTS | 1-20 | A20 | Description of the phenomenon contained in the current channel. |
| | YAT | 21-40 | A20 | Phenomenon and units of the channel. |
| | GT | 41-60 | A20 | Gage type used to record this channel. |
| | ICB | 61-65 | 15 | Calibration code: 1, if there are calibrations for this channel, or 0, if there are no calibrations for this channel. |

***The E type format can be overridden by a decimal point in the data field.**

**TABLE 1. DEFINITION OF VARIABLES ON DATA INFORMATION LINES FOR THE UNIFORM FORMAT FOR EXPERIMENTAL DATA (continued)**

| Record Number | Variable Name | Column Numbers | Format | Description |
|---|---|---|---|---|
| 3 | SBL | 66-70 | 15 | Value, in analog-to-digital counts, to bring the data to the zero baseline. |
| 4 | SS | 1-15 | E15* | Value, in electrical units or in physical units, of the top step of the calibration steps. |
| | SA,SB,SC | 16-60 | 3E15* | Quadratic calibration coefficients used to convert the raw data into electrical units. |
| | A | 61-75 | E15* | The first of three quadratic transformation coefficients used to convert the electrical units into physical units. |
| 5 | B,C | 1-30 | 2E15* | The second and third quadratic coefficients for the conversion of electrical units to physical units. |

*** The E type format can be overridden by a decimal point in the data field.**

```
31U-DF             1 3 7UP-DOWN FRONT ID 31                                    1
      2048                                                                     1
             0.         .00050                                                 1
     125.0     124.0     124.0     125.0     125.0     125.0     125.0     125.0
     124.0     125.0     125.0     125.0     125.0     124.0     125.0     126.0
     125.0     124.0     125.0     125.0     125.0     125.0     125.0     125.0
     125.0     124.0     124.0     125.0     125.0     124.0     126.0     125.0
     125.0     125.0     125.0     126.0     125.0     125.0     125.0     125.0
     124.0     125.0     125.0     126.0     125.0     125.0     124.0     125.0
     124.0     124.0     125.0     125.0     124.0     124.0     125.0     125.0
      .
      .
      .
     115.0     115.0     114.0     114.0     114.0     114.0     114.0     113.0
     113.0     112.0     113.0     113.0     114.0     112.0     112.0     112.0
     113.0     113.0     113.0     113.0     113.0     114.0     114.0     114.0
     115.0     115.0     115.0     115.0     115.0     114.0     116.0     115.0
     115.0     115.0     115.0     117.0     117.0     118.0     125.0     125.0
FILE 31U-DF:C12
```

Figure 2. An Example of an Input Format Accepted by ADTAPE4 and Other Data Analysis Programs

## TABLE 2. DEFINITION OF VARIABLES ON DATA INFORMATION LINES OF DATA FILE FORMAT 2

| Record Number | Variable Name | Column Numbers | Format | Description |
|---|---|---|---|---|
| 1 | NTS* | 1-48 | 24A2 | File description including file name, round number, channel number, total number of channels and message. These variables will be described in Table 3. |
| 2 | NPT | 1-10 | I10 | Total number of data points in this channel. |
| 3 | TIME | 1-15 | F15.8 | Time at fiducial mark. |
|  | DT | 16-30 | F15.8 | Time between data points. |

***Table 3 describes the variable NTS. NTS is made up of 24 two-column words. Words six and seven of NTS have been left blank.**

## TABLE 3. EXPLANATION OF CONTENTS OF VARIABLE NTS

| Variable Name | Index of NTS | Format | Column Numbers | Description |
|---|---|---|---|---|
| NAMEF | 1-5 | 5A2 | 1-10 | The name of the data file, which usually contains the round number from the original analog tape. |
| IRD | 8 | 12 | 15-16 | The round number* of this round in relation to other rounds in this file. |
| ICNT | 9 | 12 | 17-18 | The channel number in relation to the total number of channels in this round. |
| NCH | 10 | 12 | 19-20 | The total number of channels of this round in this file. |
| MSSG | 11-24 | 14A2 | 21-48 | An optional 28 columns that may be used for additional description of the channel. It is taken from the storage information of files stored on an HP9845. |

***Program ADTAPE3 counts the rounds that are sequentially stored on a file. This round number generally has nothing to do with the round number on the original analog tape. In most cases, this number will not be greater than one for the files converted on an HP9845.**

The first ten rows of Figure 2 are the data information records (one through three) and the first seven data lines. These are followed by three blank lines and the last five data records. The last line of Figure 2 is printed as hardcopy output, but is not stored on the disk. This line gives the name of the file that was converted.

Figure 3 is a summary of all of the channels converted to be put in a single file and lists the first five lines of each channel converted separated by three blank lines. The final line of the hardcopy output is not stored on disk. This line prints the name of the disk file on which all of this data are stored.

## III. ACTUAL TRANSFER OF DATA

Detailed instructions for the transfer of files between an HP9845 and MFA are found in Appendix A of Reference 2 which is reproduced in Appendix F for the reader's convenience. Files may be transferred in either direction, i.e., from HP to MFA and from MFA to HP.

It takes approximately 10 wall-clock minutes to transfer 1813 lines of data from HP to MFA at a rate of 4800 baud (bits per second).

## IV. READING THE FILES ON THE CDC MAINFRAMES

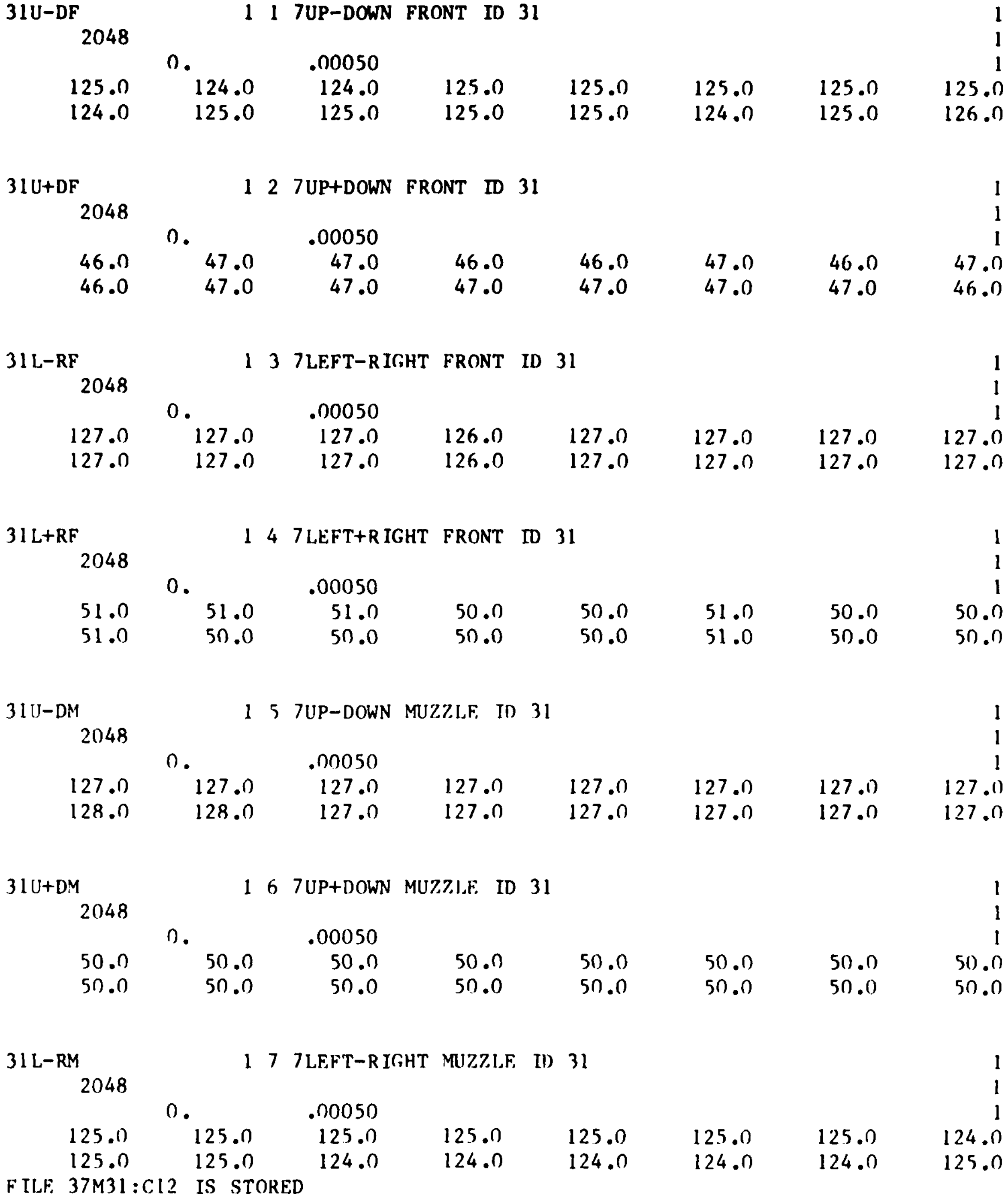
There are four short segments of job control language (JCL) and FORTRAN IV programs stored on MFA that have been written to read the ASCII files transferred from an HP9845. Two of the program files (CONHP and CVHPZ) contain the formats used to read the data files created by CNVTB1. The remaining two program files (CVHPZ3 and CVHPA) read the data files set up by CNVTB4.

The FORTRAN programs contained in files CONHP and CVHPA are combined with the JCL established to run on MFA. The JCLs designed for MFZ are stored with the FORTRAN programs in files CHVPZ and CVHPZ3. The MFZ is normally used for larger programs and faster execution time. The MFB shares files with MFA; therefore, files that are stored on MFA are accessible from MFB. MFB is used primarily for interactive graphics applications.

In order to use a data file created by program CNVTB4 in program ADTAPE4 and various other existing data analysis programs, the data must be read in using formatted input and re-stored on a file using unformatted output. This process is the function of the FORTRAN programs in files CVHPZ3 and CVHPA. The BALDPRESS version of the BROAD program accepts the data file written by CNVTB1 as it exists. There is no need for an intermediate program. For use in other programs, the formats in the FORTRAN programs in files CONHP and CVHPZ are recommended for reading the data.

All of the FORTRAN programs listed will plot each data channel separately. Therefore, you may desire to run files CONHP or CVHPZ in order to get a plot of the data as it currently exists. These programs will not change or re-save the data; they will simply verify the existence of the data in MFA by printout and plot.

Appendix C contains JCL for MFA, JCL for MFZ and the corresponding FORTRAN program designed to read data files created by program CNVTB1.



```
31U-DF            1 1 7UP-DOWN FRONT ID 31                                  1
      2048                                                                  1
             0.          .00050                                             1
     125.0     124.0     124.0     125.0     125.0     125.0     125.0     125.0
     124.0     125.0     125.0     125.0     125.0     124.0     125.0     126.0

31U+DF            1 2 7UP+DOWN FRONT ID 31                                  1
      2048                                                                  1
             0.          .00050                                             1
      46.0      47.0      47.0      46.0      46.0      47.0      46.0      47.0
      46.0      47.0      47.0      47.0      47.0      47.0      47.0      46.0

31L-RF            1 3 7LEFT-RIGHT FRONT ID 31                               1
      2048                                                                  1
             0.          .00050                                             1
     127.0     127.0     127.0     126.0     127.0     127.0     127.0     127.0
     127.0     127.0     127.0     126.0     127.0     127.0     127.0     127.0

31L+RF            1 4 7LEFT+RIGHT FRONT ID 31                               1
      2048                                                                  1
             0.          .00050                                             1
      51.0      51.0      51.0      50.0      50.0      51.0      50.0      50.0
      51.0      50.0      50.0      50.0      50.0      51.0      50.0      50.0

31U-DM            1 5 7UP-DOWN MUZZLE ID 31                                 1
      2048                                                                  1
             0.          .00050                                             1
     127.0     127.0     127.0     127.0     127.0     127.0     127.0     127.0
     128.0     128.0     127.0     127.0     127.0     127.0     127.0     127.0

31U+DM            1 6 7UP+DOWN MUZZLE ID 31                                 1
      2048                                                                  1
             0.          .00050                                             1
      50.0      50.0      50.0      50.0      50.0      50.0      50.0      50.0
      50.0      50.0      50.0      50.0      50.0      50.0      50.0      50.0

31L-RM            1 7 7LEFT-RIGHT MUZZLE ID 31                              1
      2048                                                                  1
             0.          .00050                                             1
     125.0     125.0     125.0     125.0     125.0     125.0     125.0     124.0
     125.0     125.0     124.0     124.0     124.0     124.0     124.0     125.0
FILE 37M31:C12 IS STORED
```

**Figure 3. An Example of a Summary of Multiple Data Files Converted, Consolidated and Stored on One File**

Appendix D lists JCL for MFA, JCL for MFZ and the associated FORTRAN program established to read data files converted using program CNVTB4.

The files CONHP, CVHPA, CVHPZ and CVHPZ3 are saved on MFA under UN=CAL. These files are in the public domain and may be read and changed. However, when changes are made, they must be saved under the name of the current user. They may not be re-saved under UN=CAL.

## V. TRANSFER OF DATA FILES CREATED ON CDC MAINFRAMES TO HP9845

In addition, it is often desirable to transfer data files created on CDC mainframes to an HP9845 for additional analysis. Analysis on an HP9845 is often more desirable because of faster turnaround with smaller computations. It also frees the mainframes for larger number-crunching jobs.

In order for a file to be accessible to an HP9845, it must be stored on MFA. If a file is created on MFZ, it may be stored on MFA using the SAVEPF command. The SAVEPF command follows the same parameter list as the GETPF command illustrated in Appendices C and D.

An additional method of getting a file stored on MFA is to submit the job using the SUBMIT,fn,T command. This returns the output to MFA from MFZ and provides the option of editing, printing, saving or otherwise manipulating the data before transfer. The transfer will be faster if all unnecessary information is edited out before the file is cataloged and/or transferred.

As was previously stated, the mechanics of the actual transfer of data from the CDC to an HP9845 can be found in Appendix F.

## VI. READING THE FILES ON AN HP9845

In order for the transferred data files to be used on an HP9845, they must be converted from ASCII to a numeric data type, either digital or binary. For greater efficiency in reading and for more compact storage of the data, it is more desirable to convert the data to a binary format. The program (SORTAL:C) is designed to accomplish this conversion.

The format for the data storage conforms to the format designed for use in the ANPACK[4] routines. These routines are the basic analysis packages used on the HP9845s in the IBD of the BRL. This data format is also used in numerous other analysis and plotting routines developed for use on an HP9845.

The SORTAL program converts the ASCII data, whether it is one data file stored in rows or in several data files stored in columns. The general layout of a transferred data file conforms to a computer printout in which the data are printed out one phenomena at a time in rows or several phenomena at a time printed out in columns. The listing of the SORTAL routine along with the STOSOR routine, which is used to store files on disk, can be found in Appendix E.

[4] *J.N. Walbert, "Interactive Data Acquisition, Analysis, and Presentation Software for Experimental Research Facilities in the Interior Ballistics Division at BRL," to be published.*

## VII. SUMMARY

BASIC programs CNVTB1 and CNVTB4 were designed to convert binary data stored on an HP9845 to ASCII. This conversion enables this data to be transferred to MFA. FORTRAN programs in files CONHP, CVHPA, CVHPZ and CVHPZ3 were written for the purpose of reading the transferred data and putting it in a format suitable for use by data analysis programs that reside on MFA or MFZ.

The program SORTAL was written to convert ASCII data to binary. This process enables an HP9845 to read and use data that have been transferred from MFA to an HP9845. This data can then be further analyzed and/or plotted on an HP9845 using existing software or software to be created.

# APPENDIX A

## LISTING OF PROGRAM CNVTB1

# APPENDIX A

## LISTING OF PROGRAM CNVTB1

```
10     OPTION BASE 1                      !FILE CNVTB1:C
20     !  CONVERTS DATA FROM BDAT TO ASCII FOR POTENTIAL TRANSFER TO CDC
30     SHORT Data(5000)
40     DIM Label$[64],Datal$(300)[80],Temp$(1)[80],Value$[80],Ss(10),Datas$(2100)[80]
50     INTEGER Datai(0:5000)
60     Ident$="   65"
70     Devname$="SIMULATION"
80     MASS STORAGE IS ":C12"
90     !
100    !   INPUT NUMBER OF FILES TO CONVERT AND CONSOLIDATE
110    !
120    PRINT PAGE
130    PRINT "HOW MANY FILES DO YOU WANT TO CONVERT AND CONSOLIDATE?"
140    INPUT Number
150    PRINT PAGE
160    I=1
170    Ss(1)=1
180    J1=1
190    !
200    !   SET UP SOME ASCII VALUES OF ZERO, ONE AND BLANK
210    !
220    One$="    1"
230    Zero$="    0"
240    Blank$="     "
250    Blank4$="    "
260    Point$="."
270    !
280    !   READ THE FILE TO BE CONVERTED
290    !
300    PRINT "ENTER FILE NAME"
310    INPUT Filename$
320    Filename$=Filename$&":C12"
330    PRINT "WAS THE DATA STORED FROM THE ACQUISITION (INTEGER) ROUTINE (0) OR"
340    PRINT "FROM THE ANALYSIS (SHORT) ROUTINE (1)"
350    INPUT From
360    IF From=1 THEN Ana
370    !
380    !   READ DATA STORED USING ACQUISITION ROUTINE
390    !
400    FREAD Filename$,Datai(*)
410    MAT Data=Datai
420    Data(37)=1/Data(37)
430    GOTO Next
440    !
450    !   READ DATA STORED USING ANALYSIS ROUTINE
460    !
470  Ana:FREAD Filename$,Data(*)
480    !
490    !  PRINT FILE INFORMATION AND DETERMINE IF CORRECT FILE
500    !
```

```
510 Next:Nrec=ROW(Data)
520   PRINT PAGE
530   Nwords=Nrec-37
540   Format$="#,64A"
550   IF From=1 THEN ENTER Data(1) USING Format$;Label$
560   IF From=0 THEN ENTER Datai(0) USING Format$;Label$
570   PRINT "STORAGE INFORMATION FOR FILE ";Filename$;LIN(1);Label$
580   PRINT "START=";Data(33);"STOP=";Data(34);"INC=";Data(35)
590   PRINT "DELTA TIME=";Data(37)
600   PRINT "NO. OF RECORDS = ";Data(36)
610   PRINT LIN(2)
620   PRINT "IS THIS THE CORRECT FILE?  'Y' or 'N'"
630   INPUT Right$
640   PRINT PAGE
650   IF Right$<>"Y" THEN 300
660   !
670   !  SET UP THE INFORMATION LINES WITH INFORMATION FROM THE FILE
680   !
690   FOR N=1 TO 50 STEP 10
700   Temp$(1)[N;10]=Label$[N;2]&Label$[N+2;2]&Label$[N+4;2]&Label$[N+6;2]&
Label$[N+8;2]
710   NEXT N
720   PRINTER IS 0
730   Data1$(1)[1;5]="    2"
740   Data1$(1)[6;30]=Temp$(1)[1;10]&Temp$(1)[11;10]&Temp$(1)[21;10]
750   Data1$(1)[36;10]=Temp$(1)[31;10]
760   Data1$(1)[46;10]=Temp$(1)[41;10]
770   Data1$(1)[56;5]=Ident$
780   Data1$(1)[61;5]=One$
790   Data1$(1)[66;15]=Blank$&Blank$&Zero$
800   Data1$(2)[1;10]=Devname$
810   Data1$(2)[11;5]=One$
820   Data1$(2)[16;5]=Zero$
830   Data1$(2)[21;10]=VAL$(Data(34))
840   FIXED 9
850   Data1$(2)[31;15]=Blank$&VAL$(Data(37))
860   STANDARD
870   Data1$(2)[46;15]=Blank$&Blank4$&Zero$&Point$
880   Data1$(2)[61;5]="    8"
890   Data1$(2)[66;10]="  (8F10.1)"
900   Data1$(2)[76;5]=Zero$
910   A=2
920   Stat=1
930   Nn1=21
940   Nn2=30
950   GOTO Justify
960 Back:Nn1=31
970   Nn2=45
980   Stat=2
990   A=2
1000  GOTO Justify
1010  !
1020  !   CONVERT THE NUMERICAL DATA POINTS TO ASCII
1030  !
```

```
1040 Convert:PRINT PAGE
1050  PRINT "CONVERTING DATA TO ASCII"
1060  J=6
1070  L=38
1080  Ii=INT((Nrec-37)/8)+2
1090  FOR K=1 TO 80 STEP 10
1100  Data1$(J)[K;10]=VAL$(Data(L))
1110  A=J
1120  Stat=3
1130  Nn1=K
1140  Nn2=K+9
1150  GOTO Justify
1160 Data_trans:L=L+1
1170  IF L>Nrec THEN 1340
1180  NEXT K
1190  J=J+1
1200  GOTO 1090
1210  !
1220  !   IF THE FINAL DATA LINE DOES NOT HAVE EIGHT DATA POINTS,
1230  !   FILL THE REST OF THE DATA POINTS WITH ZEROES TO MAKE 80 COLUMNS
1240  !
1250  PRINT PAGE
1260  Nleft=Nwords MOD 8
1270  IF Nleft=0 THEN 1340
1280  FOR Kk=K+10 TO 80 STEP 10
1290  Data1$(J)[Kk;10]="     0.000"
1300  NEXT Kk
1310  !
1320  !   PRINT THE SUMMARY OF THE CONVERTED FILE
1330  !
1340  PRINT PAGE
1350  Mm=J
1360  PRINTER IS 0
1370  FOR Jj=1 TO 10
1380  PRINT Data1$(Jj)
1390  NEXT Jj
1400  PRINT LIN(1)
1410  FOR Jj=Mm-4 TO Mm
1420  PRINT Data1$(Jj)
1430  NEXT Jj
1440  PRINT "FILE ";Filename$
1450  PRINT LIN(1)
1460  PRINTER IS 16
1470  GOTO Trans
1480  !
1490  !  EACH OF THE DATA POINTS IS RIGHT-JUSTIFIED IN A TEN-COLUMN FIELD
1500  !
1510 Justify:Savenn=0
1520  Dec_pt=0
1530  FOR Nn=Nn1 TO Nn2
1540  IF Data1$(A)[Nn;1]=" " THEN Savenn=Savenn+1
1550  IF Data1$(A)[Nn;1]="." THEN Dec_pt=1
1560  NEXT Nn
1570  IF Savenn=0 THEN Savenn=1
```

```
1580  Nnspan=Nn2-Nn1+1
1590  Nchar=Nnspan-Savenn
1600  FOR L1=Nn2 TO Nn2-Nchar+1 STEP -1
1610  Data1$(A)[L1;1]=Data1$(A)[L1-Savenn;1]
1620  Data1$(A)[L1-Savenn;1]=" "
1630  NEXT L1
1640  IF Stat=1 THEN 1660
1650  IF Dec_pt=0 THEN 1680
1660  ON Stat GOTO Back,Default,Data_trans
1670  STOP
1680  FOR H=Nn2-Nchar+1 TO Nn2
1690  Data1$(A)[H-2;1]=Data1$(A)[H;1]
1700  NEXT H
1710  Data1$(A)[Nn2;1]="0"
1720  Data1$(A)[Nn2-1;1]="."
1730  GOTO 1660
1740  !
1750  !  SET UP DEFAULT FILE INFORMATION FROM INPUT FILE
1760  !
1770 Default:PRINTER IS 16
1780  Data1$(3)[1;20]=Data1$(1)[6;20]
1790  Data1$(3)[21;20]=Data1$(1)[26;20]
1800  Data1$(3)[41;20]=Blank$&Blank$&Blank$&Blank$
1810  Data1$(3)[61;5]=Zero$
1820  Data1$(3)[66;5]=Zero$
1830  Data1$(3)[71;10]=Blank$&Zero$
1840  Data1$(4)[1;15]=Blank$&Blank4$&Zero$&Point$
1850  Data1$(4)[16;15]=Blank$&Blank4$&Zero$&Point$
1860  Data1$(4)[31;15]=Blank$&Blank4$&One$&Point$
1870  Data1$(4)[46;15]=Blank$&Blank4$&Zero$&Point$
1880  Data1$(4)[61;15]=Blank$&Blank4$&Zero$&Point$
1890  Data1$(4)[76;5]=Zero$
1900  Data1$(5)[1;15]=Blank$&Blank4$&One$&Point$
1910  Data1$(5)[16;15]=Blank$&Blank4$&Zero$&Point$
1920  Data1$(5)[31;50]=Blank$&Blank$&Blank$&Blank$&Blank$&Blank$&Blank$
&Blank$&Blank$&Zero$
1930  !
1940  !   PRINT THE DEFAULT VALUES DERIVED FROM THE INPUT FILE
1950  !
1960 Info:PRINT PAGE
1970  PRINT "These are the default values from your file and as otherwise"
1980  PRINT "designated. If you want to change any of them, ENTER 'C'."
1990  PRINT
2000  PRINT "FILE VERSION = ";Data1$(1)[1;5];" PROJECT NAME = ";Data1$(1)[6;30]
2010  PRINT "DATE = ";Data1$(1)[36;10];" TIME = ";Data1$(1)[46;10];" IDENT NO. = ";
Data1$(1)[56;5]
2020  PRINT "NO. OF DEVICES = ";Data1$(1)[61;5];" DEVICE NAME = ";Data1$(2)[1;10]
2030  PRINT "NO. OF CHANNELS = ";Data1$(2)[11;5];" NO. OF CAL WORDS = ";
Data1$(2)[16;5]
2040  PRINT "NO. OF DATA WORDS = ";Data1$(2)[21;10];" DELTA TIME = ";
Data1$(2)[31;15]
2050  PRINT "OFFSET TIME = ";Data1$(2)[46;15];" WORDS PER CARD AS SET BY THIS
PROGRAM =";Data1$(2)[61;5]
2060  PRINT "DATA FORMAT = ";Data1$(2)[66;10];" GAGE TYPE = ";Data1$(3)[41;20]
```

```
2070  PRINT "CHANNEL DESCRIPTION = ";Datal$(3)[1;20]
2080  PRINT "QUANTITY AND UNITS = ";Datal$(3)[21;20]
2090  PRINT "CAL CODE ( 1 IF THERE ARE CALS, 0 IF THERE ARE NO CALS )= ";
Datal$(3)[61;5]
2100  PRINT "BASELINE = ";Datal$(3)[66;5];" TOP STEP GAGE RESPONSE = ";
Datal$(4)[1;15]
2110  PRINT "QUADRATIC CALIBRATION COEFFICIENTS ";
Datal$(4)[16;15];Datal$(4)[31;15];Datal$(4)[46;15]
2120  PRINT "QUADRATIC TRANSFORMATION COEFFICIENTS ";
Datal$(4)[61;15];Datal$(5)[1;15];Datal$(5)[16;15]
2130  PRINT LIN(1)
2140  !
2150  !  HERE IS THE OPTION TO CHANGE ANY INFORMATION THAT HAS BEEN LISTED
2160  !  AND RIGHT-JUSTIFY ANY NEW INFORMATION
2170  !
2180  PRINT "DO YOU WANT TO CHANGE ANY ITEMS? ENTER 'C' TO CHANGE"
2190  BEEP
2200  INPUT Change$
2210  IF Change$<>"C" THEN Convert
2220  PRINT
2230  PRINT "IF YOU WANT TO CHANGE A VARIABLE, ENTER THE VALUE THAT YOU WANT;"
2240  PRINT "OTHERWISE, ENTER 'N'. IF YOU HAVE CHANGED THE LAST VALUE THAT YOU"
2250  PRINT "WANT TO CHANGE,  ENTER 'E'"
2260  PRINT "FILE VERSION (I5)=";Datal$(1)[1;5]
2270  INPUT Value$
2280  IF Value$="E" THEN Convert
2290  IF Value$="N" THEN 2380
2300  Datal$(1)[1;5]=Value$
2310  FOR E=5 TO 1 STEP -1
2320  IF Datal$(1)[5;1]<>" " THEN 2380
2330  FOR F=4 TO 1 STEP -1
2340  Datal$(1)[F+1;1]=Datal$(1)[F;1]
2350  NEXT F
2360  Datal$(1)[5-E+1;1]=" "
2370  NEXT E
2380  PRINT "PROJECT NAME (A30)=";Datal$(1)[6;30]
2390  INPUT Value$
2400  IF Value$="E" THEN Convert
2410  IF Value$="N" THEN 2430
2420  Datal$(1)[6;30]=Value$
2430  PRINT "DATE (A10)=";Datal$(1)[36;10]
2440  INPUT Value$
2450  IF Value$="E" THEN Convert
2460  IF Value$="N" THEN 2480
2470  Datal$(1)[36;10]=Value$
2480  PRINT "TIME (A10)=";Datal$(1)[46;10]
2490  INPUT Value$
2500  IF Value$="E" THEN Convert
2510  IF Value$="N" THEN 2530
2520  Datal$(1)[46;10]=Value$
2530  PRINT "IDENT NO. (I5)=";Datal$(1)[56;5]
2540  INPUT Value$
2550  IF Value$="E" THEN Convert
2560  IF Value$="N" THEN 2650
```

```
2570  Data1$(1)[56;5]=Value$
2580  FOR E=60 TO 56 STEP -1
2590  IF Data1$(1)[60;1]<>" " THEN 2650
2600  FOR F=59 TO 56 STEP -1
2610  Data1$(1)[F+1;1]=Data1$(1)[F;1]
2620  NEXT F
2630  Data1$(1)[60-E+56;1]=" "
2640  NEXT E
2650  PRINT "NO. OF DEVICES (I5)=";Data1$(1)[61;5]
2660  INPUT Value$
2670  IF Value$="E" THEN Convert
2680  IF Value$="N" THEN 2770
2690  Data1$(1)[61;5]=Value$
2700  FOR E=65 TO 61 STEP -1
2710  IF Data1$(1)[65;1]<>" " THEN 2770
2720  FOR F=64 TO 61 STEP -1
2730  Data1$(1)[F+1;1]=Data1$(1)[F;1]
2740  NEXT F
2750  Data1$(1)[65-E+61;1]=" "
2760  NEXT E
2770  PRINT "DEVICE NAME (A10)=";Data1$(2)[1;10]
2780  INPUT Value$
2790  IF Value$="E" THEN Convert
2800  IF Value$="N" THEN 2820
2810  Data1$(2)[1;10]=Value$
2820  PRINT "NO. OF CHANNELS (I5)=";Data1$(2)[11;5]
2830  INPUT Value$
2840  IF Value$="E" THEN Convert
2850  IF Value$="N" THEN 2940
2860  Data1$(2)[11;5]=Value$
2870  FOR E=15 TO 11 STEP -1
2880  IF Data1$(2)[15;1]<>" " THEN 2940
2890  FOR F=14 TO 11 STEP -1
2900  Data1$(2)[F+1;1]=Data1$(2)[F;1]
2910  NEXT F
2920  Data1$(2)[15-E+11;1]=" "
2930  NEXT E
2940  PRINT "NO. OF CAL WORDS (I5)=";Data1$(2)[16;5]
2950  INPUT Value$
2960  IF Value$="E" THEN Convert
2970  IF Value$="N" THEN 3060
2980  Data1$(2)[16;5]=Value$
2990  FOR E=20 TO 16 STEP -1
3000  IF Data1$(2)[20;1]<>" " THEN 3060
3010  FOR F=19 TO 16 STEP -1
3020  Data1$(2)[F+1;1]=Data1$(2)[F;1]
3030  NEXT F
3040  Data1$(2)[20-E+16;1]=" "
3050  NEXT E
3060  PRINT "NO. OF DATA WORDS (I10)=";Data1$(2)[21;10]
3070  INPUT Value$
3080  IF Value$="E" THEN Convert
3090  IF Value$="N" THEN 3180
3100  Data1$(2)[21;10]=Value$
```

```
3110  FOR E=30 TO 21 STEP -1
3120  IF Datal$(2)[30;1]<>" " THEN 3180
3130  FOR F=29 TO 21 STEP -1
3140  Datal$(2)[F+1;1]=Datal$(2)[F;1]
3150  NEXT F
3160  Datal$(3)[30-E+21;1]=" "
3170  NEXT E
3180  PRINT "DELTA TIME (E15)=";Datal$(2)[31;15]
3190  INPUT Value$
3200  IF Value$="E" THEN Convert
3210  IF Value$="N" THEN 3370
3220  Datal$(2)[31;15]=Value$
3230  Isign=0
3240  CALL Decimal(Datal$(2),31,15,Isign)
3250  IF (Datal$(2)[45;1]<>" ") AND (Isign=1) THEN 3370
3260  FOR E=45 TO 31 STEP -1
3270  IF Datal$(2)[44;1]<>" " THEN 3370
3280  Is=44
3290  IF Isign=1 THEN 3320
3300  Is=43
3310  IF (E=45) AND (Isign=0) THEN Datal$(2)[45;1]=Point$
3320  FOR F=Is TO 31 STEP -1
3330  Datal$(2)[F+1;1]=Datal$(2)[F;1]
3340  NEXT F
3350  Datal$(2)[45-E+31;1]=" "
3360  NEXT E
3370  PRINT "OFFSET TIME (E15)=";Datal$(2)[46;15]
3380  INPUT Value$
3390  IF Value$="E" THEN Convert
3400  IF Value$="N" THEN 3560
3410  Datal$(2)[46;15]=Value$
3420  Isign=0
3430  CALL Decimal(Datal$(2),46,15,Isign)
3440  IF (Datal$(2)[60;1]<>" ") AND (Isign=1) THEN 3560
3450  FOR E=60 TO 46 STEP -1
3460  IF Datal$(2)[59;1]<>" " THEN 3560
3470  Is=59
3480  IF Isign=1 THEN 3510
3490  Is=58
3500  IF (E=60) AND (Isign=0) THEN Datal$(2)[60;1]=Point$
3510  FOR F=Is TO 46 STEP -1
3520  Datal$(2)[F+1;1]=Datal$(2)[F;1]
3530  NEXT F
3540  Datal$(2)[60-E+46;1]=" "
3550  NEXT E
3560  PRINT "WORDS PER CARD IMAGE (set in program) (I5)=";Datal$(2)[61;5]
3570  INPUT Value$
3580  IF Value$="E" THEN Convert
3590  IF Value$="N" THEN 3680
3600  Datal$(2)[61;5]=Value$
3610  FOR E=65 TO 61 STEP -1
3620  IF Datal$(2)[65;1]<>" " THEN 3680
3630  FOR F=64 TO 61 STEP -1
3640  Datal$(2)[F+1;1]=Datal$(2)[F;1]
```

```
3650  NEXT F
3660  Data1$(2)[65-E+61;1]=" "
3670  NEXT E
3680  PRINT "DATA FORMAT(set in program) (A10)=";Data1$(2)[66;10]
3690  INPUT Value$
3700  IF Value$="E" THEN Convert
3710  IF Value$="N" THEN 3730
3720  Data1$(2)[66;10]=Value$
3730  PRINT "CHANNEL DESCRIPTION (A20)=";Data1$(3)[1;20]
3740  INPUT Value$
3750  IF Value$="E" THEN Convert
3760  IF Value$="N" THEN 3780
3770  Data1$(3)[1;20]=Value$
3780  PRINT "QUANTITY AND UNITS (A20)=";Data1$(3)[21;20]
3790  PRINT "USE A BLANK INSTEAD OF A COMMA TO SEPARATE THE UNITS FROM THE
QUANTITY"
3800  PRINT "EXAMPLE: PRESSURE MPA"
3810  INPUT Value$
3820  IF Value$="E" THEN Convert
3830  IF Value$="N" THEN 3850
3840  Data1$(3)[21;20]=Value$
3850  PRINT "GAGE TYPE (A20)=";Data1$(3)[41;20]
3860  INPUT Value$
3870  IF Value$="E" THEN Convert
3880  IF Value$="N" THEN 3900
3890  Data1$(3)[41;20]=Value$
3900  PRINT "CAL CODE (1 if there are cals, 0 if there are no cals)(I5)=";
Data1$(3)[61;5]
3910  INPUT Value$
3920  IF Value$="E" THEN Convert
3930  IF Value$="N" THEN 4020
3940  Data1$(3)[61;5]=Value$
3950  FOR E=65 TO 61 STEP -1
3960  IF Data1$(3)[65;1]<>" " THEN 4020
3970  FOR F=64 TO 61 STEP -1
3980  Data1$(3)[F+1;1]=Data1$(3)[F;1]
3990  NEXT F
4000  Data1$(3)[65-E+61;1]=" "
4010  NEXT E
4020  PRINT "BASELINE (I5)=";Data1$(3)[66;5]
4030  INPUT Value$
4040  IF Value$="E" THEN Convert
4050  IF Value$="N" THEN 4140
4060  Data1$(3)[66;5]=Value$
4070  FOR E=70 TO 66 STEP -1
4080  IF Data1$(3)[70;1]<>" " THEN 4140
4090  FOR F=69 TO 66 STEP -1
4100  Data1$(3)[F+1;1]=Data1$(3)[F;1]
4110  NEXT F
4120  Data1$(3)[70-E+66;1]=" "
4130  NEXT E
4140  PRINT "TOP STEP GAGE RESPONSE (E15)=";Data1$(4)[1;15]
4150  INPUT Value$
4160  IF Value$="E" THEN Convert
```

```
4170  IF Value$="N" THEN 4330
4180  Data1$(4)[1;15]=Value$
4190  Isign=0
4200  CALL Decimal(Data1$(4),1,15,Isign)
4210  IF (Data1$(4)[15;1]<>" ") AND (Isign=1) THEN 4330
4220  FOR E=15 TO 1 STEP -1
4230  IF Data1$(4)[14;1]<>" " THEN 4330
4240  Is=14
4250  IF Isign=1 THEN 4280
4260  Is=13
4270  IF (E=15) AND (Isign=0) THEN Data1$(4)[15;1]=Point$
4280  FOR F=Is TO 1 STEP -1
4290  Data1$(4)[F+1;1]=Data1$(4)[F;1]
4300  NEXT F
4310  Data1$(4)[15-E+1;1]=" "
4320  NEXT E
4330  PRINT "QUADRATIC CALIBRATION COEFFICIENTS (3E15)"
4340  PRINT "A";Data1$(4)[16;15]
4350  INPUT Value$
4360  IF Value$="E" THEN Convert
4370  IF Value$="N" THEN 4530
4380  Data1$(4)[16;15]=Value$
4390  Isign=0
4400  CALL Decimal(Data1$(4),16,15,Isign)
4410  IF (Data1$(4)[30;1]<>" ") AND (Isign=1) THEN 4530
4420  FOR E=30 TO 16 STEP -1
4430  IF Data1$(4)[29;1]<>" " THEN 4530
4440  Is=29
4450  IF Isign=1 THEN 4480
4460  Is=28
4470  IF (E=30) AND (Isign=0) THEN Data1$(4)[30;1]=Point$
4480  FOR F=Is TO 16 STEP -1
4490  Data1$(4)[F+1;1]=Data1$(4)[F;1]
4500  NEXT F
4510  Data1$(4)[30-E+16;1]=" "
4520  NEXT E
4530  PRINT "B";Data1$(4)[31;15]
4540  INPUT Value$
4550  IF Value$="E" THEN Convert
4560  IF Value$="N" THEN 4720
4570  Data1$(4)[31;15]=Value$
4580  Isign=0
4590  CALL Decimal(Data1$(4),31,15,Isign)
4600  IF (Data1$(4)[45;1]<>" ") AND (Isign=1) THEN 4720
4610  FOR E=45 TO 31 STEP -1
4620  IF Data1$(4)[44;1]<>" " THEN 4720
4630  Is=44
4640  IF Isign=1 THEN 4670
4650  Is=43
4660  IF (E=45) AND (Isign=0) THEN Data1$(4)[45;1]=Point$
4670  FOR F=Is TO 31 STEP -1
4680  Data1$(4)[F+1;1]=Data1$(4)[F;1]
4690  NEXT F
4700  Data1$(4)[45-E+31;1]=" "
```

```
4710  NEXT E
4720  PRINT "C";Data1$(4)[46;15]
4730  INPUT Value$
4740  IF Value$="E" THEN Convert
4750  IF Value$="N" THEN 4910
4760  Data1$(4)[46;15]=Value$
4770  Isign=0
4780  CALL Decimal(Data1$(4),46,15,Isign)
4790  IF (Data1$(4)[60;1]<>" ") AND (Isign=1) THEN 4910
4800  FOR E=60 TO 46 STEP -1
4810  IF Data1$(4)[59;1]<>" " THEN 4910
4820  Is=59
4830  IF Isign=1 THEN 4860
4840  Is=58
4850  IF (E=60) AND (Isign=0) THEN Data1$(4)[60;1]=Point$
4860  FOR F=Is TO 46 STEP -1
4870  Data1$(4)[F+1;1]=Data1$(4)[F;1]
4880  NEXT F
4890  Data1$(4)[60-E+46;1]=" "
4900  NEXT E
4910  PRINT "QUADRATIC TRANSFORMATION COEFFICIENTS (3E15)"
4920  PRINT "A";Data1$(4)[61;15]
4930  INPUT Value$
4940  IF Value$="E" THEN Convert
4950  IF Value$="N" THEN 5110
4960  Data1$(4)[61;15]=Value$
4970  Isign=0
4980  CALL Decimal(Data1$(4),61,15,Isign)
4990  IF (Data1$(4)[75;1]<>" ") AND (Isign=1) THEN 5110
5000  FOR E=75 TO 61 STEP -1
5010  IF Data1$(4)[74;1]<>" " THEN 5110
5020  Is=74
5030  IF Isign=1 THEN 5060
5040  Is=73
5050  IF (E=75) AND (Isign=0) THEN Data1$(4)[75;1]=Point$
5060  FOR F=Is TO 61 STEP -1
5070  Data1$(4)[F+1;1]=Data1$(4)[F;1]
5080  NEXT F
5090  Data1$(4)[75-E+61;1]=" "
5100  NEXT E
5110  PRINT "B";Data1$(5)[1;15]
5120  INPUT Value$
5130  IF Value$="E" THEN Convert
5140  IF Value$="N" THEN 5300
5150  Data1$(5)[1;15]=Value$
5160  Isign=0
5170  CALL Decimal(Data1$(5),1,15,Isign)
5180  IF (Data1$(5)[14;1]<>" ") AND (Isign=1) THEN 5300
5190  FOR E=15 TO 1 STEP -1
5200  IF Data1$(5)[14;1]<>" " THEN 5300
5210  Is=14
5220  IF Isign=1 THEN 5250
5230  Is=13
5240  IF (E=15) AND (Isign=0) THEN Data1$(5)[15;1]=Point$
```

```
5250  FOR F=Is TO 1 STEP -1
5260  Data1$(5)[F+1;1]=Data1$(5)[F;1]
5270  NEXT F
5280  Data1$(5)[15-E+1;1]=" "
5290  NEXT E
5300  PRINT "C";Data1$(5)[16;15]
5310  INPUT Value$
5320  IF Value$="E" THEN Convert
5330  IF Value$="N" THEN Convert
5340  Data1$(5)[16;15]=Value$
5350  Isign=0
5360  CALL Decimal(Data1$(5),16,15,Isign)
5370  IF (Data1$(5)[30;1]<>" ") AND (Isign=1) THEN 5380
5380  FOR E=30 TO 16 STEP -1
5390  IF Data1$(5)[29;1]<>" " THEN Convert
5400  Is=29
5410  IF Isign=1 THEN 5440
5420  Is=28
5430  IF (E=30) AND (Isign=0) THEN Data1$(5)[30;1]=Point$
5440  FOR F=Is TO 16 STEP -1
5450  Data1$(5)[F+1;1]=Data1$(5)[F;1]
5460  NEXT F
5470  Data1$(5)[30-E+16;1]=" "
5480  NEXT E
5490  GOTO Convert
5500  !
5510  !   CONSOLIDATION OF CONVERTED FILES INTO ONE FILE
5520  !
5530 Trans:FOR K=1 TO Mm
5540  Datas$(I)=Data1$(K)
5550  I=I+1
5560  NEXT K
5570  IF J1<>Number THEN Ss(J1+1)=I
5580  PRINT "FILE ";Filename$;" IS DONE"
5590  IF J1=Number THEN 5650
5600  J1=J1+1
5610  GOTO 300
5620  !
5630  !  INPUT THE NAME OF THE FILE TO STORE CONSOLIDATED DATA
5640  !
5650  PRINT "ENTER THE NAME OF FILE TO STORE"
5660  INPUT Filestore$
5670  Filestore$=Filestore$&":C12"
5680  !
5690  !   DETERMINE OPTIMUM FILE SIZE
5700  !
5710  Mm=I-1
5720  REDIM Datas$(Mm)
5730  Size=INT((4*Mm+84*Mm)/256)+1
5740  PRINT "Size = ";Size
5750  !
5760  !   CREATE FILE AND STORE DATA
5770  !
5780  CREATE Filestore$,Size
```

```
5790  ASSIGN #3 TO Filestore$
5800  PRINT #3;Datas$(*)
5810  ASSIGN * TO #3
5820  PRINT "STORAGE COMPLETE"
5830  !
5840  !   PRINT SUMMARY OF ALL FILES CONSOLIDATED AND STORED
5850  !
5860  PRINTER IS 0
5870  FOR Mi=1 TO Number
5880  J=Ss(Mi)
5890  FOR M=J TO J+6
5900  PRINT Datas$(M)
5910  NEXT M
5920  IF Mi<>Number THEN PRINT LIN(1)
5930  NEXT Mi
5940  PRINT "FILE ";Filestore$;" IS STORED"
5950  PRINT LIN(2)
5960  PRINTER IS 16
5970  !
5980  !   OPTION TO CONVERT AND/OR CONSOLIDATE ANOTHER SET OF FILES
5990  !
6000  INPUT "DO YOU WANT TO CONVERT AND CONSOLIDATE ANOTHER SET OF FILES? Y or
N",Anymore$
6010  IF Anymore$="Y" THEN 120
6020  PRINT "PROGRAM TERMINATED"
6030  STOP
6040  END
6050  !
6060  !    SUBROUTINE TO CHECK FOR DECIMAL POINT IN A NUMBER
6070  !
6080  SUB Decimal(Data$,Is,Nl,Isign)
6090  OPTION BASE 1
6100  FOR I=Is TO Nl+Is-1
6110  IF Data$[I;1]="." THEN Isign=1
6120  NEXT I
6130  SUBEND
```

## APPENDIX B

## LISTING OF PROGRAM CNVTB4

## APPENDIX B

## LISTING OF PROGRAM CNVTB4

```
10     OPTION BASE 1                          !FILE CNVTB4:C
20     !  CONVERTS DATA FROM BDAT TO ASCII FOR POTENTIAL TRANSFER TO CDC
30     COM SHORT Data(5000)
40     COM Label$[64],Data1$(300)[80],Temp$(1)[80],Value$[80],Msg$[24],
Datas$(2100)[80]
50     COM Nrec,Filename$,Nlines,I,Filename1$,Ss(10),Number
60     MASS STORAGE IS ":C12"
70     !
80     !    INPUT THE DESIRED PROGRAM OPTION
90     !
100    PRINT "DO YOU WANT TO CONSOLIDATE FILES THAT HAVE ALREADY BEEN
CONVERTED"
110    PRINT "TO ALPHANUMERIC?   Y or N"
120    INPUT Consol$
130    IF Consol$<>"Y" THEN Conv
140    Consolop=1
150    GOTO Combin
160  Conv:Consolop=2
170    !
180    !    INPUT NUMBER OF FILES TO CONVERT AND CONSOLIDATE
190    !
200    PRINT PAGE
210    PRINT "HOW MANY FILES DO YOU WANT TO CONVERT AND CONSOLIDATE?"
220    INPUT Number
230    PRINT PAGE
240    I=1
250    Ss(1)=1
260    J1=1
270    !
280    !  SET UP SOME ASCII VALUES OF ZERO, ONE AND BLANK
290    !
300    One2$=" 1"
310    One6$="     1"
320    One$="    1"
330    Zero$="    0"
340    Blank$="     "
350    Blank5$="     "
360    !
370    !   READ THE FILE TO BE CONVERTED
380    !
390    REDIM Data1$(300),Datas$(2100)
400    CALL Readin(1,0)
410    !
420    !   PUT THE INFORMATION FROM VARIABLE LABEL$ INTO TEMP$
430    !
440    FOR N=1 TO 50 STEP 10
450    Temp$(1)[N;10]=Label$[N;2]&Label$[N+2;2]&Label$[N+4;2]&Label$[N+6;2]&
Label$[N+8;2]
460    NEXT N
470    PRINTER IS 0
```

```
480   !
490   !  SET UP THE INFORMATION LINES WITH INFORMATION FROM THE FILE
500   !
510   Data1$(1)[1;10]=Filename1$
520   Data1$(1)[11;4]=Blank$
530   Data1$(1)[15;6]=One6$
540   Data1$(1)[21;2]=One2$
550   Data1$(1)[23;2]=One2$
560   Data1$(1)[25;10]=Temp$(1)[1;10]
570   Data1$(1)[35;10]=Temp$(1)[11;10]
580   Data1$(1)[45;4]=Temp$(1)[21;4]
590   Data1$(1)[49;6]=Blank5$&" "
600   Data1$(1)[55;15]=Blank5$&Blank5$&Blank5$
610   Data1$(1)[70;11]=Blank5$&One6$
620   Data1$(2)[1;10]=VAL$(Data(34))
630   Data1$(2)[11;20]=Blank5$&Blank5$&Blank5$&Blank5$
640   Data1$(2)[31;20]=Blank5$&Blank5$&Blank5$&Blank5$
650   Data1$(2)[51;20]=Blank5$&Blank5$&Blank5$&Blank5$
660   Data1$(2)[71;10]=Blank5$&One$
670   Data1$(3)[1;15]=Blank5$&Blank$&Zero$&"."
680   FIXED 9
690   Data1$(3)[16;15]=Blank$&VAL$(Data(37))
700   STANDARD
710   Data1$(3)[31;20]=Blank5$&Blank5$&Blank5$&Blank5$
720   Data1$(3)[51;20]=Blank5$&Blank5$&Blank5$&Blank5$
730   Data1$(3)[71;10]=Blank5$&One$
740   A=2
750   Stat=1
760   Nn1=1
770   Nn2=10
780   GOTO Justify
790 Back:Nn1=16
800   Nn2=30
810   Stat=2
820   A=3
830   GOTO Justify
840   !
850   !   CONVERT THE NUMERICAL DATA POINTS TO ASCII
860   !
870 Convert:PRINT PAGE
880   PRINT "CONVERTING DATA TO ASCII"
890   J=4
900   L=38
910    I1=INT((Nrec-37)/8)+2
920   FOR K=1 TO 80 STEP 10
930   Data1$(J)[K;10]=VAL$(Data(L))
940   A=J
950   Stat=3
960   Nn1=K
970   Nn2=K+9
980   GOTO Justify
990 Data_trans:L=L+1
1000   IF L>Nrec THEN 1080
1010   NEXT K
```

```
1020  J=J+1
1030  GOTO 920
1040  !
1050  !   IF THE FINAL DATA LINE DOES NOT HAVE EIGHT DATA POINTS,
1060  !   FILL THE REST OF THE DATA POINTS WITH ZEROES TO MAKE 80 COLUMNS
1070  !
1080  PRINT PAGE
1090  Nleft=Nwords MOD 8
1100   IF Nleft=0 THEN 1140
1110  FOR Kk=K+10 TO 80 STEP 10
1120  Data1$(J)[Kk;10]="     0.000"
1130  NEXT Kk
1140  Mm=J
1150  Nlines=Mm
1160  REDIM Data(5000)
1170  !
1180  !   PRINT THE SUMMARY OF THE CONVERTED FILE
1190  !
1200  PRINTER IS 0
1210  FOR O=1 TO 10
1220  PRINT Data1$(O)
1230  NEXT O
1240  PRINT LIN(2)
1250  FOR O=Mm-4 TO Mm
1260  PRINT Data1$(O)
1270  NEXT O
1280  PRINT "FILE ";Filename$
1290  PRINT LIN(2)
1300  PRINTER IS 16
1310  GOTO Trans
1320 From:PRINT PAGE
1330  !
1340  !  EACH OF THE DATA POINTS IS RIGHT-JUSTIFIED IN A TEN-COLUMN FIELD
1350  !
1360 Justify:Savenn=0
1370  Dec_pt=0
1380  FOR Nn=Nn1 TO Nn2
1390   IF Data1$(A)[Nn;1]=" " THEN Savenn=Savenn+1
1400   IF Data1$(A)[Nn;1]="." THEN Dec_pt=1
1410  NEXT Nn
1420   IF Savenn=0 THEN Savenn=1
1430  Nnspan=Nn2-Nn1+1
1440  Nchar=Nnspan-Savenn
1450  FOR L1=Nn2 TO Nn2-Nchar+1 STEP -1
1460  Data1$(A)[L1;1]=Data1$(A)[L1-Savenn;1]
1470  Data1$(A)[L1-Savenn;1]=" "
1480  NEXT L1
1490   IF Stat=1 THEN 1510
1500   IF Dec_pt=0 THEN 1530
1510  ON Stat GOTO Back,Info,Data_trans
1520  STOP
1530  FOR H=Nn2-Nchar+1 TO Nn2
1540  Data1$(A)[H-2;1]=Data1$(A)[H;1]
1550  NEXT H
```

```
1560  Data1$(A)[Nn2;1]="0"
1570  Data1$(A)[Nn2-1;1]="."
1580  GOTO 1510
1590  !
1600  !    PRINT THE DEFAULT VALUES DERIVED FROM THE INPUT FILE
1610  !
1620 Info:PRINTER IS 16
1630  PRINT PAGE
1640  PRINT "These are the default values from your file and as otherwise"
1650  PRINT "designated. If you want to change any of them, ENTER 'C'."
1660  PRINT
1670  PRINT "FILE NAME = ";Data1$(1)[1;10];" FILE NUMBER = ";Data1$(1)[15;6]
1680  PRINT "CHANNEL NO. = ";Data1$(1)[21;2];" TOTAL NO. OF CHANNELS = ";
Data1$(1)[23;2]
1690  PRINT "MESSAGE = ";Data1$(1)[25;24]
1700  PRINT "NO. OF DATA PTS = ";Data1$(2)[1;10]
1710  PRINT "START TIME = ";Data1$(3)[1;15];" DELTA TIME = ";Data1$(3)[16;15]
1720  PRINT LIN(1)
1730  !
1740  ! HERE IS THE OPTION TO CHANGE ANY INFORMATION THAT HAS BEEN LISTED
1750  ! AND RIGHT-JUSTIFY ANY NEW INFORMATION
1760  !
1770  PRINT "DO YOU WANT TO CHANGE ANY ITEMS? ENTER 'C' TO CHANGE"
1780  BEEP
1790  INPUT Change$
1800  IF Change$<>"C" THEN Convert
1810  PRINT
1820  PRINT "IF YOU WANT TO CHANGE A VARIABLE, ENTER THE VALUE THAT YOU WANT;"
1830  PRINT "OTHERWISE, ENTER 'N'. IF YOU HAVE CHANGED THE LAST VALUE THAT
YOU"
1840  PRINT "WANT TO CHANGE,  ENTER 'E'"
1850  PRINT "FILE NAME (5A2) =";Data1$(1)[1;10]
1860  INPUT Value$
1870  IF Value$="E" THEN Convert
1880  IF Value$="N" THEN 1970
1890  Data1$(1)[1;10]=Value$
1900  FOR E=10 TO 1 STEP -1
1910  IF Data1$(1)[10;1]<>" " THEN 1970
1920  FOR F=9 TO 1 STEP -1
1930  Data1$(1)[F+1;1]=Data1$(1)[F;1]
1940  NEXT F
1950  Data1$(1)[10-E+1;1]=" "
1960  NEXT E
1970  PRINT "FILE NUMBER (3A2)=";Data1$(1)[15;6]
1980  INPUT Value$
1990  IF Value$="E" THEN Convert
2000  IF Value$="N" THEN 2090
2010  Data1$(1)[15;6]=Value$
2020  FOR E=20 TO 15 STEP -1
2030  IF Data1$(1)[20;1]<>" " THEN 2090
2040  FOR F=19 TO 15 STEP -1
2050  Data1$(1)[F+1;1]=Data1$(1)[F;1]
2060  NEXT F
2070  Data1$(1)[20-E+15;1]=" "
```

```
2080  NEXT E
2090  PRINT "CHANNEL NO. (A2)=";Data1$(1)[21;2]
2100  INPUT Value$
2110  IF Value$="E" THEN Convert
2120  IF Value$="N" THEN 2170
2130  Data1$(1)[21;2]=Value$
2140  IF Data1$(1)[22;1]<>" " THEN 2170
2150  Data1$(1)[22;1]=Data1$(1)[21;1]
2160  Data1$(1)[21;1]=" "
2170  PRINT "TOTAL NO. OF CHANNELS (A2) = ";Data1$(1)[23;2]
2180  INPUT Value$
2190  IF Value$="E" THEN Convert
2200  IF Value$="N" THEN 2250
2210  Data1$(1)[23;2]=Value$
2220  IF Data1$(1)[24;1]<>" " THEN 2250
2230  Data1$(1)[24;1]=Data1$(1)[23;1]
2240  Data1$(1)[23;1]=" "
2250  PRINT "MESSAGE (12A2) =";Data1$(1)[25;24]
2260  INPUT Value$
2270  IF Value$="E" THEN Convert
2280  IF Value$="N" THEN 2300
2290  Data1$(1)[25;24]=Value$
2300  PRINT "NO. OF DATA PTS (I10)=";Data1$(2)[1;10]
2310  INPUT Value$
2320  IF Value$="E" THEN Convert
2330  IF Value$="N" THEN 2420
2340  Data1$(2)[1;10]=Value$
2350  FOR E=10 TO 1 STEP -1
2360  IF Data1$(2)[10;1]<>" " THEN 2420
2370  FOR F=9 TO 1 STEP -1
2380  Data1$(2)[F+1;1]=Data1$(2)[F;1]
2390  NEXT F
2400  Data1$(2)[10-E+1;1]=" "
2410  NEXT E
2420  PRINT "START TIME (E15.6)=";Data1$(3)[1;15]
2430  INPUT Value$
2440  IF Value$="E" THEN Convert
2450  IF Value$="N" THEN 2610
2460  Data1$(3)[1;15]=Value$
2470  Isign=0
2480  CALL Decimal(Data1$(3),1,15,Isign)
2490  IF (Data1$(3)[15;1]<>" ") AND (Isign=1) THEN 2610
2500  FOR E=15 TO 1 STEP -1
2510  IF Data1$(3)[14;1]<>" " THEN 2610
2520  Is=14
2530  IF Isign=1 THEN 2560
2540  Is=13
2550  IF (E=15) AND (Isign=0) THEN Data1$(3)[15;1]="."
2560  FOR F=Is TO 1 STEP -1
2570  Data1$(3)[F+1;1]=Data1$(3)[F;1]
2580  NEXT F
2590  Data1$(3)[15-E+1;1]=" "
2600  NEXT E
2610  PRINT "DELTA TIME (E15.6)=";Data1$(3)[16;15]
```

```
2620  INPUT Value$
2630  IF (Value$="E") OR (Value$="N") THEN Convert
2640  Data1$(3)[16;15]=Value$
2650  Isign=0
2660  CALL Decimal(Data1$(3),16,15,Isign)
2670  IF (Data1$(3)[30;1]<>" ") AND (Isign=1) THEN 2680
2680  FOR E=30 TO 16 STEP -1
2690  IF Data1$(3)[30;1]<>" " THEN Convert
2700  Is=29
2710  IF Isign=1 THEN 2740
2720  Is=28
2730  IF (E=30) AND (Isign=0) THEN Data1$(3)[30;1]="."
2740  FOR F=Is TO 16 STEP -1
2750  Data1$(3)[F+1;1]=Data1$(3)[F;1]
2760  NEXT F
2770  Data1$(3)[30-E+16;1]=" "
2780  NEXT E
2790  GOTO Convert
2800  !
2810  !  CONSOLIDATION OF CONVERTED FILES INTO ONE FILE
2820  !
2830 Combin:PRINT PAGE
2840  REDIM Datas$(2100),Data1$(300)
2850  INPUT "HOW MANY FILES DO YOU WANT TO CONSOLIDATE?",Number
2860  I=1
2870  Ss(1)=1
2880  !
2890  !    READ THE FILES TO BE CONSOLIDATED
2900  !
2910  FOR J=1 TO Number
2920  CALL Readin(2,J)
2930  !
2940  !  STORE THE CONVERTED FILE INTO ARRAY DATAS$
2950  !
2960 Trans:FOR K=1 TO Nlines
2970  Datas$(I)=Data1$(K)
2980  I=I+1
2990  NEXT K
3000  IF J1<>Number THEN Ss(J1+1)=I
3010  PRINT "FILE ";Filename$;" IS DONE"
3020  IF Consolop=1 THEN 3060
3030  IF J1=Number THEN 3100
3040  J1=J1+1
3050  GOTO 390
3060  NEXT J
3070  !
3080  !   STORE THE CONVERTED AND CONSOLIDATED FILE
3090  !
3100  PRINT "READY TO ENTER STORE"
3110  CALL Store(2,I-1)
3120  !
3130  !  OPTION TO CONSOLIDATE ANOTHER SET OF FILES OR END PROGRAM
3140  !
3150  IF Consolop=2 THEN 3200
```

```
3160  INPUT "DO YOU WANT TO CONSOLIDATE ANOTHER SET OF FILES? Y or N",Again$
3170  PRINT PAGE
3180  IF Again$="Y" THEN Combin
3190  GOTO 3260
3200  INPUT "DO YOU WANT TO CONVERT AND CONSOLIDATE ANOTHER SET OF FILES? Y or
N",Again$
3210  IF Again$="Y" THEN Conv
3220  INPUT "DO YOU WANT TO CONSOLIDATE A SET OF ALREADY CONVERTED FILES? Y or
N",Change$
3230  IF Change$="N" THEN 3260
3240  Consolop=1
3250  GOTO Combin
3260  PRINT "PROGRAM STOPPED"
3270  STOP
3280  END
3290  !
3300  !  SUBROUTINE TO READ DATA FROM ACQUISITION ROUTINE, ANALYSIS ROUTINE
3310  !  OR PREVIOUSLY STORED AS ASCII
3320  !
3330  SUB Readin(In,J)
3340  !
3350  OPTION BASE 1
3360  COM SHORT Data(5000)
3370  COM Label$[64],Data1$(300)[80],Temp$(1)[80],Value$[80],Msg$[24],
Datas$(2100)[80]
3380  COM Nrec,Filename$,Nlines,I,Filename1$,Ss(10),Number
3390  INTEGER Datai(0:5000)
3400  IF In=2 THEN Alpha
3410  !
3420  !   INPUT THE NAME OF THE FILE TO BE CONVERTED AND HOW IT WAS STORED
3430  !
3440  INPUT "ENTER NAME OF FILE TO CONVERT",Filename$
3450  Filename1$=Filename$
3460  Filename$=Filename$&":C12"
3470  PRINT "WAS THE DATA STORED FROM THE ACQUISITION ROUTINE (0) OR"
3480  PRINT "FROM THE ANALYSIS ROUTINE(1)"
3490  INPUT From
3500  IF From=1 THEN Ana
3510  !
3520  !  READ DATA STORED USING ACQUISITION ROUTINE
3530  !
3540  FREAD Filename$,Datai(*)
3550  MAT Data=Datai
3560  Data(37)=1/Data(37)
3570  GOTO Next
3580  !
3590  !   READ DATA STORED USING ANALYSIS ROUTINE
3600  !
3610 Ana:FREAD Filename$,Data(*)
3620  !
3630  !   PRINT FILE INFORMATION AND DETERMINE IF CORRECT FILE
3640  !
3650 Next:Nrec=ROW(Data)
3660  PRINT PAGE
```

```
3670  Nwords=Nrec-37
3680  Format$="#,64A"
3690  IF From=1 THEN ENTER Data(1) USING Format$;Label$
3700  IF From=0 THEN ENTER Datai(0) USING Format$;Label$
3710  PRINT "STORAGE INFORMATION FOR FILE ";Filename$;LIN(1);Label$
3720  PRINT "START=";Data(33);"STOP=";Data(34);"INC=";Data(35)
3730  PRINT "DELTA TIME=";Data(37)
3740  PRINT "NO. OF RECORDS = ";Data(36)
3750  PRINT LIN(2)
3760  PRINT "IS THIS THE CORRECT FILE?  Y′ or  N′"
3770  INPUT Right$
3780  PRINT PAGE
3790  IF Right$<>"Y" THEN 3440
3800  GOTO Quit
3810  !
3820  !  READ FILE THAT HAS ALREADY BEEN CONVERTED TO ASCII
3830  !  FOR PURPOSES OF CONSOLIDATION
3840  !
3850 Alpha:PRINT "WHAT IS THE NAME OF FILE NUMBER ";J
3860  INPUT Filename$
3870  Filename$=Filename$&":C12"
3880  ASSIGN #1 TO Filename$
3890  READ #1;Data1$(1)
3900  READ #1;Data1$(2)
3910  Nwords=VAL(Data1$(2)[1;10])
3920  Nlines=Nwords/8+3
3930  ASSIGN * TO #1
3940  REDIM Data1$(Nlines)
3950  ASSIGN #1 TO Filename$
3960  READ #1;Data1$(*)
3970  ASSIGN * TO #1
3980  REDIM Data1$(300)
3990 Quit:SUBEND
4000  END
4010  !
4020  !   SUBROUTINE TO STORE FILE THAT HAS BEEN CONSOLIDATED
4030  !
4040  SUB Store(In,Mm)
4050  !
4060  OPTION BASE 1
4070  COM SHORT Data(5000)
4080  COM Label$[64],Data1$(300)[80],Temp$(1)[80],Value$[80],Msg$[24],
Datas$(2100)[80]
4090  COM Nrec,Filename$,Nlines,I,Filename1$,Ss(10),Number
4100  REDIM Datas$(Mm)
4110  !
4120  !   DETERMINE OPTIMUM FILE SIZE
4130  !
4140  Size=INT((4*Mm+84*Mm)/256)+1
4150  PRINT "Size = ";Size
4160  !
4170  !  INPUT THE NAME OF THE FILE TO STORE CONSOLIDATED FILE
4180  !
4190  PRINT "ENTER THE NAME OF THE FILE TO STORE (6 CHAR MAX.)"
```

```
4200  INPUT Filestore$
4210  Filestore$=Filestore$&":C12"
4220  !
4230  !   CREATE FILE AND STORE DATA
4240  !
4250  CREATE Filestore$,Size
4260  ASSIGN #3 TO Filestore$
4270  PRINT #3;Datas$(*)
4280  ASSIGN * TO #3
4290  PRINT "STORAGE COMPLETE"
4300  !
4310  !   PRINT SUMMARY OF ALL FILES CONSOLIDATED AND STORED
4320  !
4330  PRINTER IS 0
4340  FOR Mi=1 TO Number
4350  J=Ss(Mi)
4360  FOR M=J TO J+4
4370  PRINT Datas$(M)
4380  NEXT M
4390  IF Mi<>Number THEN PRINT LIN(2)
4400  NEXT Mi
4410  PRINT "FILE ";Filestore$;" IS STORED"
4420  PRINT LIN(2)
4430  PRINTER IS 16
4440  SUBEND
4450  END
4460  !
4470  !    SUBROUTINE TO CHECK FOR DECIMAL POINT IN A NUMBER
4480  !
4490  SUB Decimal(Data$,Is,N1,Isign)
4500  OPTION BASE 1
4510  FOR I=Is TO N1+Is-1
4520  IF Data$[I;1]="." THEN Isign=1
4530  NEXT I
4540  SUBEND
4550  END
```

# APPENDIX C

## LISTING OF FORTRAN PROGRAM IN FILES CONHP AND CVHPZ ALONG WITH ASSOCIATED JOB CONTROL LANGUAGES FOR MFA AND MFZ

# APPENDIX C

## LISTING OF FORTRAN PROGRAM IN FILES CONHP AND CVHPZ ALONG WITH ASSOCIATED JOB CONTROL LANGUAGES FOR MFA AND MFZ

I. Job Control Language for MFA

```
Usern.
USER,username,password.
CHARGE,account,project number.
BEGIN,ATTACH,PLOTLIB.
GET,TAPE1=filename.

     where filename = name of file in which data are stored

FTN.
LGO.
BEGIN,PLOT,CALCOMP,TAPE13.
```

II. Job Control Language for MFZ

```
Usern,STMFZ.
ACCOUNT,account number.
BEGIN,ATTACH,PLOTLIB.
REQUEST,TAPE13,*PF.
GETPF,TAPE1,pfn,UN=usn,UP=pwd,ST=MFA,CH=acctno,PN=projno.

     where pfn = name of file from which the data are read
           usn = username
           pwd = user password
           acctno = computer account number
           projno = project number


FILE,TAPE1,CM=YES,MRL=81,RT=F.
FTN,I=INPUT,R=3,SL.
MAP,OFF.
LGO.
CATALOG,TAPE13,pfn,ID=userid.


     where pfn = name of file in which plot is to be stored


BEGIN,PLOT,CALCOMP,TAPE13.
```

## III. FORTRAN PROGRAM

```
      PROGRAM CONVR( INPUT,OUTPUT,TAPE1,TAPE5=INPUT,TAPE6=OUTPUT)
      DIMENSION D(5000),TITL(3),NTS(2),YAT(2),GT(2)
      READ(1,1000) IFV,TITL,DATE,TIME,IROUND,IADC
      READ(1,1020) IA,NCH,ITZ,NBD,DELT,OFFTIM,NUMBER,IRDFMT
      NCHA=NCH
      DO 200 I=1,NCHA
      IF(K.GT.1) READ(1,1000) IFV,TITL,DATE,TIME,IROUND,IADC
      IF(K.EQ.1) WRITE(6,1010) IFV,TITL,DATE,TIME,IROUND,IADC
      IF(K.GT.1) WRITE(6,1110) IFV,TITL,DATE,TIME,IROUND,IADC
      IF(K.GT.1) READ(1,1020) IA,NCH,ITZ,NBD,DELT,OFFTIM,NUMBER,IRDFMT
      WRITE(6,1030) IA,NCH,ITZ,NBD,DELT,OFFTIM,NUMBER,IRDFMT
      READ(1,1040) NTS,YAT,GT,ICB,SBL
      WRITE(6,1050) NTS,YAT,GT,ICB,SBL
      READ(1,1060) SS,SA,SB,SC,A
      WRITE(6,1070) SS
      WRITE(6,1080) SA,SB,SC
      READ(1,1090) B,C
      WRITE(6,1100) A,B,C
      DO 100 J=1,NBD,NUMBER
      JJ=J+NUMBER-1
      IF(JJ.GT.NBD)JJ=NBD
      READ(1,IRDFMT) (D(I),I=J,JJ)
      IF(J.LT.25)WRITE(6,IRDFMT)(D(I),I=J,JJ)
  100 CONTINUE
      WRITE(6,1120)
      CALL PLOT(D,NBD,NTS,DELT,K)
  200 CONTINUE
      STOP
 1000 FORMAT( I5,5A10,2I5)
 1010 FORMAT('1FILE VERSION =',I5,2X,'PROJECT NAME =',3A10/' DATE =',A10
     *       ,2X,'TIME = ',A10,2X,'IDENT =',I5,2X,'NO. OF DEVICES =',I5)
 1020 FORMAT(A10,2I5,I10,2F15.4,I5,A10)
 1030 FORMAT(' DEVICE NAME = ',A10,2X,'NO. OF CHANNELS =',I5,2X,
     *        'NO. OF CAL WORDS =',I5/' NO. OF DATA WORDS =',I10,2X,
     *        'DELT =',F15.4/' OFFSET TIME =',F15.4,2X,'WORDS PER CARD ='
     *       ,I5,2X,'DATA FORMAT =',A10)
 1040 FORMAT(6A10,2I5)
 1050 FORMAT(' CHANNEL DESCRIPTION =',2A10,2X,'QUANTITY AND UNITS =',
     *        2A10/' GAGE TYPE =',2A10,2X,'CAL CODE =',I5,2X,'BASELINE ='
     *       ,I5)
 1060 FORMAT(5F15.4)
 1070 FORMAT(' TOP STEP GAGE RESPONSE =',F15.4)
 1080 FORMAT(' CALIBRATION COEFFICIENTS'/3(F15.4,2X))
 1090 FORMAT(2F15.4)
 1100 FORMAT(' CALIBRATION TRANSFORMATION COEFFICIENTS'/3(F15.4,2X))
 1110 FORMAT(' FILE VERSION =',I5,2X,'PROJECT NAME =',3A10/' DATE =',A10
     *       ,2X,'TIME = ',A10,2X,'IDENT =',I5,2X,'NO. OF DEVICES =',I5)
 1120 FORMAT(//)
      END
      SUBROUTINE PLOT(D,NPT,NTS,DELT,K)
      DIMENSION LABEL(4),ITITLE(3),D(5000),NTS(2),COUNT(5000)
      IF(K.GT.1) GO TO 100
      XB=1.75
      YB=.4
      XS=500.
      DX=500.
      XMI=.0
      XAX=NPT/500+1
      XMA=XS*XAX
      YAX=4.
      XPAGE=XAX+2.2
      YPAGE=4.75
      FACT=1.
      IUNIT=13
      LABEL(1)=10H CAL 390
      LABEL(2)=10H 6126
      LABEL(3)=10H37MM DATA
      LABEL(4)=10HFROM HP45
      MODE=1
      CALL PLTBEG(XPAGE,YPAGE,FACT,IUNIT,LABEL)
      YMI=0.
      YMA=300.
      DY=50.
      YS=(YMA-YMI)/YAX
  100 CALL PLTSCA(XB,YB,XMI,YMI,XS,YS)
      CALL PLTWND(XMI,XMA,YMI,YMA)
      CALL PLTAXS(DX,DY,XMI,XMA,YMI,YMA,4)
      CALL LABELA(DX,DY,XMI,XMA,YMI,YMA,1.,1.)
      TX=XMI
      TY=YMA+.05*YS
      CHT=.1
      TIC=DX*DELT*1000.
      TIM=DELT*FLOAT(NPT)*1000.
      ENCODE(22,2000,ITITLE)(NTS(I),I=1,2)
 2000 FORMAT(1H ,2A10,1H>)
      CALL PLTSYM(CHT,ITITLE,0.,TX,TY)
      TX=XMI-XS*1.55
      TY=YMA-YS*.3
      ITITLE(1)=10HTIME, MS>
      CALL PLTSYM(CHT,ITITLE,0.,TX,TY)
      TY=YMA-YS*.45
      ENCODE(18,3000,ITITLE)TIM
 3000 FORMAT(8H /FRAME=,F9.3,1H>)
      CALL PLTSYM(CHT,ITITLE,0.,TX,TY)
      TY=YMA-YS*.6
      ENCODE(16,4000,ITITLE)TIC
 4000 FORMAT(8H /TIC=   ,F7.3,1H>)
      CALL PLTSYM(CHT,ITITLE,0.,TX,TY)
      DO 200 I=1,NPT
      COUNT(I)=I
  200 CONTINUE
      CALL PLTDTS(MODE,0,COUNT(1),D(1),NPT,0)
      CALL PLTPGE
      RETURN
      END
```

# APPENDIX D

## LISTING OF FORTRAN PROGRAM IN FILES CVHPA AND CVHPZ3 ALONG WITH ASSOCIATED JOB CONTROL LANGUAGES FOR MFA AND MFZ

# APPENDIX D

## LISTING OF FORTRAN PROGRAM IN FILES CVHPA AND CVHPZ3 ALONG WITH ASSOCIATED JOB CONTROL LANGUAGES FOR MFA AND MFZ

I. Job Control Language for MFA

```
Usern.
USER,username, password.
CHARGE,account,project number.
BEGIN,ATTACH,PLOTLIB.
GET,TAPE1=filename.

     where filename = name of file in which data are stored

FTN.
LGO.
SAVE,TAPE2=pfn.

     where pfn = name of file in which to store reformatted data

BEGIN,PLOT,CALCOMP,TAPE13.
```

II. Job Control Language for MFZ

```
Usern,STMFZ.
ACCOUNT,account number.
BEGIN,ATTACH,PLOTLIB.
REQUEST,TAPE13,*PF.
GETPF,TAPE1,pfn,UN=usn,UP=pwd,ST=MFA,CH=acctno,PN=projno.

     where pfn = name of file from which the data are read
           usn = username
           pwd = user password
           acctno = computer account number
           projno = project number

FILE,TAPE1,CM=YES,MRL=81,RT=F.
FTN,I=INPUT,R=3,SL.
MAP,OFF.
LGO.
CATALOG,TAPE2,pfn,ID=userid.

     where pfn = name of file in which to store reformatted data

CATALOG,TAPE13,pfn,ID=userid.

     where pfn = name of file in which plot is to be stored

BEGIN,PLOT,CALCOMP,TAPE13.
```

## III. FORTRAN Program

```
      PROGRAM CONVR(INPUT,OUTPUT,TAPE1,TAPE5=INPUT,TAPE6=OUTPUT,TAPE2,
     *TAPE13)
      DIMENSION D(5000),NTS(24)
      READ(1,1000) NTS
      DECODE(2,1080,NTS(12))NCHS
      DO 200 K=1,NCHS
      IF(K.GT.1) READ(1,1000)NTS
      IF(K.EQ.1) WRITE(6,1010) NTS
      IF(K.GT.1) WRITE(6,1100) NTS
      WRITE(2) NTS
      READ(1,1020) NPT
      WRITE(6,1030) NPT
      WRITE(2) NPT
      READ(1,1040) TZERO,DELT
      WRITE(6,1050) TZERO,DELT
      WRITE(2) TZERO,DELT
      DO 100 J=1,NPT,8
      JJ=J+7
      IF(J+7.GT.NPT)JJ=NPT
      READ(1,1060) (D(I),I=J,JJ)
      IF(J.LT.20) WRITE(6,1070) (D(I),I=J,JJ)
  100 CONTINUE
      WRITE(6,1090)
      WRITE(2) (D(I),I=1,NPT)
      CALL PLOT (D,NPT,NTS,DELT,K)
  200 CONTINUE
      STOP
 1010 FORMAT(1H1,'NTS = ',24A2)
 1000 FORMAT(24A2)
 1030 FORMAT(' NO. OF DATA POINTS = ',I10)
 1020 FORMAT(I10)
 1050 FORMAT(' TZERO = ',F15.8,' DELT = ',F15.8)
 1040 FORMAT(2F15.6)
 1060 FORMAT(8F10.0)
 1070 FORMAT(8F10.4)
 1080 FORMAT(I2)
 1090 FORMAT(//)
 1100 FORMAT(' NTS = ',24A2)
      END
      SUBROUTINE PLOT(D,NPT,NTS,DELT,K)
      DIMENSION LABEL(4),ITITLE(3),D(5000),NTS(24),COUNT(5000)
      IF(K.GT.1) GO TO 100
      XB=1.75
      YB=.4
      XS=500.
      DX=500.
      XMI=.0
      XAX=NPT/500+1
      XMA=XS*XAX
      YAX=4.
      XPAGE=XAX+2.2
      YPAGE=4.75
      FACT=1.
      IUNIT=13
      LABEL(1)=10H CAL 390
      LABEL(2)=10H 6126
      LABEL(3)=10H37MM DATA
      LABEL(4)=10HFROM HP45
      MODE=1
      CALL PLTBEG(XPAGE,YPAGE,FACT,IUNIT,LABEL)
      YMI=0.
      YMA=300.
      DY=50.
      YS=(YMA-YMI)/YAX
  100 CALL PLTSCA(XB,YB,XMI,YMI,XS,YS)
      CALL PLTWND(XMI,XMA,YMI,YMA)
      CALL PLTAXS(DX,DY,XMI,XMA,YMI,YMA,4)
      CALL LABELA(DX,DY,XMI,XMA,YMI,YMA,1.,1.)
      ENCODE(30,1000,ITITLE) (NTS(I),I=6,10)
 1000 FORMAT(3H ID,5A2,8H    PLOT,8X,1H>)
      TX=XMI
      TY=YMA+.05*YS
      CHT=.1
      CALL PLTSYM(CHT,ITITLE(1),0.,TX,TY)
      TIC=DX*DELT*1000.
      TIM=DELT*FLOAT(NPT)*1000.
      TY=YMA+.20*YS
      ENCODE(12,2000,ITITLE)(NTS(I),I=1,5)
 2000 FORMAT(1H ,5A2,1H>)
      CALL PLTSYM(CHT,ITITLE,0.,TX,TY)
      TX=XMI-XS*1.55
      TY=YMA-YS*.3
      ITITLE(1)=10HTIME, MS>
      CALL PLTSYM(CHT,ITITLE,0.,TX,TY)
      TY=YMA-YS*.45
      ENCODE(18,3000,ITITLE)TIM
 3000 FORMAT(8H /FRAME=,F9.3,1H>)
      CALL PLTSYM(CHT,ITITLE,0.,TX,TY)
      TY=YMA-YS*.6
      ENCODE(16,4000,ITITLE)TIC
 4000 FORMAT(8H /TIC=  ,F7.3,1H>)
      CALL PLTSYM(CHT,ITITLE,0.,TX,TY)
      DO 200 I=1,NPT
      COUNT(I)=I
  200 CONTINUE
      CALL PLTDTS(MODE,0,COUNT(1),D(1),NPT,0)
      CALL PLTPGE
      RETURN
      END
```

## APPENDIX E

## LISTING OF PROGRAM SORTAL

# APPENDIX E

## LISTING OF PROGRAM SORTAL

```
10    OPTION BASE 1          !   FILE SORTAL:C
20    ! CONVERTS DATA TRANSFERRED FROM MFA TO HP DISC IN CARD IMAGE
30    COM SHORT Array(1900),Ar(10),Delt,Ne
40    COM Label$[64],Prog$,Filestore$
50    COM INTEGER Ii,Maxfile,Index,Iit
60    COM SHORT A1(1800),A2(1800),A3(1800),A4(1800),A5(1800)
70    COM SHORT A6(1800),A7(1800)
80    DIM A$(402)[256],B$[256],C$[256],S$[1],Temp$[256]
90    !
100   !       SET ARRAYS TO ZERO
110   !
120    S$=" "
130    MAT A1=ZER
140    MAT A2=ZER
150    MAT A3=ZER
160    MAT A4=ZER
170    MAT A5=ZER
180    MAT A6=ZER
190    MAT A7=ZER
200    !
210    !     ENTER NUMBER OF DATA FILES, HOW FILES ARE STORED AND DELTA TIME
220    !
230   PRINT "ENTER THE NUMBER OF FILES IN WHICH DATA IS STORED"
240   INPUT Totalfile
250   I=2
260   Fileno=1
270   PRINT "IS DATA IN SEVERAL ARRAYS IN COLUMNS (C) OR IN ONE ARRAY"
280   PRINT "ARRANGED IN ROWS (R)"
290   INPUT Arrange$
300   PRINT PAGE
310   PRINT "ENTER THE NUMBER OF VARIABLES PER DATA LINE TO BE CONVERTED"
320   PRINT " MAX=7 IF IN COLUMNS OR MAX=8 IF IN ROWS"
330   INPUT Maxfile
340   PRINT "ARE ALL VARIABLES STORED ON 1 OR 2 LINES, ENTER NUMBER"
350   INPUT Lines
360   PRINT "ENTER THE TIME BETWEEN DATA POINTS (DELTA TIME)"
370   INPUT Delt
380   !
390   !    ENTER NAME OF FILE TO BE CONVERTED AND NUMBER OF DATA LINES
400   !
410   MASS STORAGE IS ":C12"
420   PRINT "ENTER NAME OF NEXT FILE"
430   BEEP
440   INPUT Filename$
450   Filename$=Filename$&":C12"
460   PRINT "ENTER NUMBER OF DATA LINES"
470   INPUT Imax
480   REDIM A$(Imax)
490   !
500   !   READ FILE
```

```
510   !
520   PRINT PAGE
530   ASSIGN #1 TO Filename$
540   READ #1;A$(*)
550   !
560   !    ENTER LINE AND COLUMN NUMBER TO START CONVERSION
570   !
580   PRINT "ENTER STARTING LINE NUMBER"
590   INPUT Ilineno
600   PRINT "ENTER STARTING COLUMN NUMBER"
610   INPUT Icolno
620   PRINT PAGE
630   PRINT "DATA CONVERSION IN PROGRESS"
640   !
650   !      CONVERTING DATA
660   !
670   FOR J=Ilineno TO Imax STEP Lines
680   B$=A$(J)
690   Is=0
700   M=1
710   It=Icolno
720   L1=LEN(A$(J))
730   L2=0
740   IF Lines=2 THEN L2=LEN(A$(J+1))
750   FOR L=1 TO Maxfile
760   K=0
770   IF (L=Maxfile) AND (Lines=2) THEN 860
780   IF It>L1 THEN 920
790   IF (B$[It;1]=S$) AND (Is>0) THEN 1020
800   IF (Is=0) AND (B$[It;1]=S$) THEN 840
810   IF Is=0 THEN Saveit=It
820   Is=Is+1
830   K=K+1
840   It=It+1
850   GOTO 780
860   Itemp=1
870   IF It<=87 THEN Temp$[Itemp;1]=B$[It;1]
880   Itemp=Itemp+1
890   IF It=L1 THEN 920
900   It=It+1
910   GOTO 870
920   It=8
930   IF J=Imax THEN 720
940   B$=A$(J+1)
950   IF It>L2 THEN 1000
960   IF B$[It;1]<>S$ THEN Temp$[Itemp;1]=B$[It;1]
970   Itemp=Itemp+1
980   It=It+1
990   GOTO 950
1000  Ar(L)=VAL(Temp$[1;Itemp+1])
1010  GOTO 1030
1020  Ar(L)=VAL(B$[Saveit;K])
1030  It=It+1
1040  Is=0
```

```
1050  NEXT L
1060  !
1070  !     STORE CONVERTED DATA INTO FILES TO BE STORED
1080  !
1090  IF Arrange$="R" THEN 1190
1100  IF Maxfile>=1 THEN A1(I)=Ar(1)
1110  IF Maxfile>=2 THEN A2(I)=Ar(2)
1120  IF Maxfile>=3 THEN A3(I)=Ar(3)
1130  IF Maxfile>=4 THEN A4(I)=Ar(4)
1140  IF Maxfile>=5 THEN A5(I)=Ar(5)
1150  IF Maxfile>=6 THEN A6(I)=Ar(6)
1160  IF Maxfile>=7 THEN A7(I)=Ar(7)
1170  I=I+1
1180  GOTO 1230
1190  FOR Ia=1 TO Maxfile
1200  A1(I)=Ar(Ia)
1210  I=I+1
1220  NEXT Ia
1230  NEXT J
1240  Ii=I-1
1250  PRINT PAGE
1260  IF Fileno<Totalfile THEN 1290
1270  IF Fileno=Totalfile THEN PRINT "CONVERSION COMPLETE"
1280  GOTO Store_data
1290  Fileno=Fileno+1
1300  GOTO 420
1310 Store_data:Index=1
1320  Ino=Ii+37
1330  Ne=1
1340  Iit=Ii+37
1350 Name:           !
1360  ON Index GOTO 1370,1390,1410,1430,1450,1470,1490
1370  MAT Array=A1
1380  GOTO 1530
1390  MAT Array=A2
1400  GOTO 1530
1410  MAT Array=A3
1420  GOTO 1530
1430  MAT Array=A4
1440  GOTO 1530
1450  MAT Array=A5
1460  GOTO 1530
1470  MAT Array=A6
1480  GOTO 1530
1490  MAT Array=A7
1500  !
1510  !    LOAD FILE STOSOR FOR DATA STORAGE
1520  !
1530  Prog$="SORTAL:C12"
1540  ASSIGN #3 TO "STOSOR:C"
1550  PRINT #3,7;"70    SHORT Data("&VAL$(Iit)&")"
1560  PRINT #3,8;"80    SHORT Data1("&VAL$(Iit)&")"
1570  ASSIGN #3 TO *
1580  GET "STOSOR:C12",10,Begin
```

```
1590 Back_again: PRINT "FILE  ";Index;"  STORED"
1600  IF Index=Maxfile THEN 1640
1610  IF Arrange$="R" THEN 1640
1620  Index=Index+1
1630  GOTO Name
1640  PRINT "ALL ARRAYS IN THIS PROGRAM STORED"
1650  STOP
1660  END
```

```
10 Begin:  OPTION BASE 1  ! FILE STOSOR:C12
20   COM SHORT Array(1900),Ar(10),Delt,Ne
30   COM Label$[64],Prog$,Filestore$
40   COM INTEGER Ii,Maxfile,Index,Iit
50   COM SHORT A1(1800),A2(1800),A3(1800),A4(1800),A5(1800)
60   COM SHORT A6(1800),A7(1800)
70    SHORT Data(388)
80    SHORT Data1(388)
90   M=38
100   FOR K=1 TO Ii-1
110   Data(M)=Array(K)
120   M=M+1
130   NEXT K
140    Ij=Ii+37
150   PRINT "ENTER THE NAME OF THE FILE TO BE CREATED (6 CHAR. MAX.):"
160    INPUT Filename$
170   Filename$=Filename$&":C12"
180   Size=INT(Ij*4/256)+INT(Ij*4/65536)+3
190   FCREATE Filename$,Size
200   PRINT "ENTER THE LABEL INFORMATION (64 CHAR. MAX.):"
210   LINPUT Label$
220   Data(33)=1
230   Data(34)=Ii
240   Data(35)=1
250   Data(36)=Ii+37
260   Data(37)=Delt
270   Format$="#,64A"
280   PRINTER IS 0
290   PRINT LIN(1)
300   PRINT "STORAGE INFORMATION FOR FILE ";Filename$;LIN(1);Label$
310   PRINT "START=";Data(33);"STOP=";Data(34);"INC=";Data(35)
320   PRINT "DELTA TIME=";Data(37)
330   PRINT "NO. OF RECORDS = ";Data(36)
340   PRINT LIN(1)
350   PRINTER IS 16
360   MAT Data1=Data
370   OUTPUT Data1(1) USING Format$;Label$
380   FPRINT Filename$,Data1(*)
390   LOAD Prog$,Back_again
400   END
```

# APPENDIX F

## INPUT TO CYBER MFA VIA MODEM LINK ON HP9845

## APPENDIX F

1. Preparation.

a. Turn on HP9845 and 7906 disc drive.

b. When disc drive is ready,

LOAD "ANPACK:D12",10

EXECUTE

c. When menu appears,

K3 (Data Communications)

d. Choose data link.

1 CONT for HP1000 or 2 CONT for CDC

e. Specify HP9845 file.

When "terminal ready" message appears, indicating the terminal emulator program is loaded, then

K13 (1) To get into edit mode

(2) It is not necessary to change any info on the first line. If you are going to the CDC and you do not want your password displayed on screen, you may want to change ECHO to ON by using → to space over and use STEP key to change to 'ON'.

STORE - To store data communication information line

(3) This line needs to be changed to the name of file and its size on the HP9845.

Change TEST:TI5 to filename:C12

Change SIZE=010 to # of records you want

STORE - To store file info line

(4) Will get a message that edit mode has been exited.

(5) If this is a new file, then it must be created. If it is not, skip this step.

SHIFT K12 To create file on HP9845
Answer "Y" to creation question. Message will appear on screen when creation is completed.

f. CONT Sends prompt to CDC or HP1000. Wait for LOGIN messages.

2. From CDC to HP9845.

a. LOGIN to CDC

b Repeat I.E.(1) through I.E.(4), changing first edit line back to ECHO OFF if you turned it ON.

c. /GET,A=Pfn. CR = CONT

/ASSIGN,TT,B. CR

/COPYSBF,A,B. No CR !

SHIFT K15 to turn record on

CONT

d. When finished recording,

SHIFT K15 to turn record off

e. Log off CDC

/BYE

f. SHIFT K4 to disconnect from CDC;

now back under control of HP9845.

3. From HP9845 to CDC

a. LOGIN to CDC

b. Repeat I.E.(1) through I.E.(4), changing first edit line back to ECHO OFF if you turned it ON.

c. Terminal definition defaults are set to emulate a Tektronix 4014 with a page width of 80 columns. If you need to set the terminal definition (TC parameter) to another terminal or the page width to a different size, then enter

/TRMDEF,TC= ,PW= .

See the NOS IAF Manual for parameter values. For most data transfers from the HP9845, the default parameters will be good and this step can be skipped.

d. Data transfer procedure:

NEW,LFN1
TEXT
'Enter Text Mode' message returned by CDC

SHIFT K14
"N" to handshake message from the HP9845
Data being transferred will be displayed on screen.
When all data has been transferred,

CONTROL T then CONT = CR

This terminates the input in TEXT mode and gets you back to READY mode in IAF.

e. Check the data transfer:

/REWIND,LFN1

/LIST,F=LFN1

f. To save the file:

/SAVE,LFN1=PFN/PW=______,M=______,CT=______.

g. Disconnect procedure:

/BYE

SHIFT K4

4. From HP9845 to HP1000.

a. Log in to HP1000.

b. Repeat steps I.E.(1) to I.E.(4), changing file info line to correct file name and file size.

c. :ST,1,NAMR - No CR = CONT

SHIFT K14 Upload

"N" to handshake question

CONT

d. When finished transferring to HP1000,

DO NOT SHIFT K14 as this will cause great problems.

e. Be careful not to hit

SHIFT K15 as this will write all kinds of error messages over your file on the HP9845.

f. Log off HP1000

:EX,SP

g. SHIFT   K4   to disconnect.

5. From HP1000 to HP9845.

a. Log in to HP1000.

b. Repeat steps I.E.(1) to I.E.(4), changing file info line to correct file name and file size.

c. ST, NAMR,1   No   CR   =   CONT

SHIFT   K14   to turn record on

CONT   to start recording

d. When finished recording,

SHIFT   K14   to turn record off

e. Log off HP1000

:EX,SP

f. SHIFT   K4   to disconnect

## DISTRIBUTION LIST

No of Copies — Organization

12 Administrator
Defense Technical Info Center
ATTN: DTIC-DDA
Cameron Station
Alexandria, VA 22314

1 Commander
US Army Materiel Development
and Readiness Command
ATTN: DRCDMD-ST
5001 Eisenhower Avenue
Alexandria, VA 22333

1 Commander
US Army BMD Advanced
Technology Center
ATTN: BMDATC-M, P. Boyd
P.O. Box 1500
Huntsville, AL 35804

1 Commander
US Army Materiel Development
and Readiness Command
ATTN: DRCLDC, T. Shirata
5001 Eisenhower Avenue
Alexandria, VA 22333

1 Commander
USA ARRADCOM
ATTN: DRDAR-LC, J.T. Frasier
Dover, NJ 07801

2 Commander
USA ARRADCOM
ATTN: DRDAR-TSS
Dover, NJ 07801

4 Commander
USA ARRADCOM
ATTN: DRDAR-LCU, E. Barrieres
DRDAR-LCU-M, D. Robertson
DRDAR-LCU-M, M. Weinstock
DRDAR-LCA, C. Larson
Dover, NJ 07801

No. of Copies — Organization

1 Commander
USA ARRADCOM
ATTN: DRDAR-TDC
Dover, NJ 07801

4 Commander
USA ARRADCOM
ATTN: DRDAR-LCA, B. Knutelski
DRDAR-LCR-R, E.H. Moore
DRDAR-LCS, J. Gregorits
DRDAR-LCS-D, K. Rubin
Dover, NJ 07801

8 Commander
USA ARRADCOM
ATTN: DRDAR-SCM
DRDAR-SCM, E. Bloore
DRDAR-SCM, J. Mulherin
DRDAR-SCS, B. Brodman
DRDAR-SCS, T. Hung
DRDAR-SCA, S. Jacobson
DRDAR-SCA, W. Gadomski
DRDAR-SCA, E. Malatesta
Dover, NJ 07801

3 Commander
USA ARRADCOM
ATTN: DRDAR-LCA, W. Williver
DRDAR-LCA, S. Bernstein
DRDAR-LCA, G. Demitrack
Dover, NJ 07801

4 Commander
USA ARRADCOM
ATTN: DRDAR-LCA, S. Yim
DRDAR-LCA, L. Rosendorf
DRDAR-LCA, S.H. Chu
DRDAR-LCW, R. Wrenn
Dover, NJ 07801

1 Director
USA ARRADCOM
Benet Weapons Laboratory
ATTN: DRDAR-LCB-TL
Watervliet, NY 12189

DISTRIBUTION LIST

| No of Copies | Organization |
|---|---|
| 2 | Director<br>USA ARRADCOM<br>Benet Weapons Laboratory<br>ATTN: DRDAR-LCB, T. Simkins<br>DRDAR-LCB, J. Zweig<br>Watervliet, NY 12189 |
| 2 | Commander<br>USA ARRADCOM<br>ATTN: Product Assurance Directorate<br>DRDAR-QA<br>Dover, NJ 07801 |
| 1 | Commander<br>USA ARRADCOM<br>ATTN: L. Goldsmith<br>Dover, NJ 07801 |
| 1 | Commander<br>US Army Rock Island Arsenal<br>ATTN: DRDAR-TSE-SW, R. Radkiewicz<br>Rock Island, IL 61299 |
| 1 | Commander<br>US Army Armament Materiel Readiness Command<br>ATTN: DRSAR-LEP-L<br>Rock Island, IL 61299 |
| 1 | Commander<br>US Army Aviation Research and Development Command<br>ATTN: DRDAV-E<br>4300 Goodfellow Blvd.<br>St. Louis, MO 63120 |
| 1 | Director<br>US Army Air Mobility Research and Development Laboratory<br>Ames Research Center<br>Moffett Field, CA 94035 |
| 1 | Commander<br>US Army Communicatios Rsch. and Development Command<br>ATTN: DRSEL-ATDD<br>Fort Monmouth, NJ 07703 |
| 1 | Commander<br>US Army Electronics Research and Development Command<br>Technical Support Activity<br>ATTN: DELSD-L<br>Fort Monmouth, NJ 07703 |
| 3 | Commander<br>US Army Harry Diamond Laboratories<br>ATTN: DELHD-I-TR, H.D. Curchak<br>DELHD-I-TR, H. Davis<br>DELHD-S-OE-ES, B. Banner<br>2800 Powder Mill Road<br>Adelphi, MD 20783 |
| 1 | Commander<br>US Army Harry Diamond Laboratories<br>ATTN: DELHD-TA-L<br>2800 Powder Mill Road<br>Adelphi, MD 20783 |
| 1 | Commander<br>US Army Missile Command<br>ATTN: DRSMI-R<br>Redstone Arsenal, AL 35898 |
| 1 | Commander<br>US Army Missile Command<br>ATTN: DRSMI-YDL<br>Redstone Arsenal, AL 35898 |
| 1 | Commander<br>US Army Tank Automotive Command<br>ATTN: DRSTA-TSL<br>Warren, MI 48090 |
| 2 | Commander<br>US Army Jefferson Proving Ground<br>ATTN: STEJP-TD-O, A. Tilley<br>STEJP-TD-E, J. Toomey<br>Madison, IN 47251 |

## DISTRIBUTION LIST

| No of Copies | Organization |
|---|---|
| 1 | Director<br>US Army TRADOC Systems<br>Analysis Activity<br>ATTN: ATAA-SL<br>White Sands Missile Range<br>NM 88002 |
| 2 | Commander<br>US Army Yuma Proving Ground<br>ATTN: STEYP-MTW, R. Torp<br>G. Stullenbarger<br>Yuma, AZ 85364 |
| 1 | Commander<br>US Army Research Office<br>ATTN: E. Saibel<br>P.O. Box 12211<br>Research Triangle Park<br>NC 27709 |
| 3 | Commander<br>US Army Research Office<br>P.O. Box 12211<br>ATTN: Technical Director<br>Engineering Division<br>Metallurgy & Materials<br>Division<br>Research Triangle Park, NC<br>27709 |
| 1 | Project Manager<br>Nuclear Munitions<br>ATTN: DRCPM-NUC<br>Dover, NJ 07801 |
| 1 | Project Manager<br>Tank Main Armament Systems<br>ATTN: DRCPM-TMA<br>Dover, NJ 07801 |
| 1 | Project Manager<br>Division Air Defense Gun<br>ATTN: DRCPM-ADG<br>Dover, NJ 07801 |
| 3 | Project Manager<br>Cannon Artillery Weapons<br>System<br>ATTN: DRCPM-CAWS<br>Dover, NJ 07801 |
| 1 | Product Manager for 30mm<br>Ammo.<br>ATTN: DRCPM-AAH-30mm<br>Dover, NJ 07801 |
| 1 | Product Manager<br>M110E2 Weapon System, DARCOM<br>ATTN: DRCPM-M110E2<br>Rock Island, IL 61299 |
| 2 | Commandant<br>US Army Infantry School<br>ATTN: ATSH-CD-CSO-OR<br>Fort Benning, GA 31905 |
| 2 | Commander<br>US Army Materials and<br>Mechanics Research Center<br>ATTN: J. Mescall<br>Tech. Library<br>Watertown, MA 02172 |
| 1 | Commander<br>Naval Sea Systems Command<br>(SEA-03513)<br>ATTN: L. Pasiuk<br>Washington, DC 20360 |
| 1 | Commander<br>Naval Explosive Ordnance<br>Disposal Facility<br>ATTN: Lib Div<br>Indian Head, MD 20640 |
| 1 | Superintendent<br>Naval Postgraduate School<br>ATTN: Dir of Lib<br>Monterey, CA 93940 |
| 1 | Commander<br>Naval Surface Weapons Center<br>ATTN: G-13, W.D. Ralph<br>Dahlgren, VA 22448 |

## DISTRIBUTION LIST

No of Copies — Organization

5 Commander
Naval Surface Weapons Center
ATTN: Code G-33, T. Tschirn
Code N-43, J.J. Yagla
L. Anderson
G. Soo Hoo
Code TX, W.G. Soper
Dahlgren, VA 22448

2 Commander
Naval Weapons Center
ATTN: Code 3835, R. Sewell
Code 3431, Tech Lib
China Lake, CA 93555

2 Commander
US Naval Weapons Center
ATTN: Code 608, R. Derr
Code 4505, C. Thelen
China Lake, CA 93555

2 Commander
Naval Ordnance Station
ATTN: Code 5034,
Ch. Irish, Jr.
T.C. Smith
Indian Head, MD 20640

2 AFATL (DLA)
Gun and Rocket Division
Gun Test Branch
AD3246 TEST W/TETFG
ATTN: W. Dittrich; DLJM
D. Davis; DLDL
Eglin AFB, FL 32542

1 Southwest Research Institute
ATTN: P. Cox
8500 Culebra Road
San Antonio, TX 78228

1 AFELM, The Rand Corporation
ATTN: Library-D
1700 Main Street
Santa Monica, CA 90406

1 AFWL/SUL
Kirtland AFB, NM 87117

No. of Copies — Organization

Aberdeen Proving Ground

Dir, USAMSAA
ATTN: DRXSY-D
DRXSY-MP, H. Cohen
Cdr, USATECOM
ATTN: DRSTE-TO-F
Dir, USACSL, Bldg. E3516, EA
ATTN: DRDAR-CLB-PA
DRDAR-CLN
DRDAR-CLJ-L
Dir, USAMTD
ATTN: Mr. Dykstra
Mr. Fasig
Mr. Walton

## USER EVALUATION OF REPORT

Please take a few minutes to answer the questions below; tear out this sheet, fold as indicated, staple or tape closed, and place in the mail. Your comments will provide us with information for improving future reports.

1. BRL Report Number ______________________________

2. Does this report satisfy a need? (Comment on purpose, related project, or other area of interest for which report will be used.)

________________________________________

________________________________________

________________________________________

3. How, specifically, is the report being used? (Information source, design data or procedure, management procedure, source of ideas, etc.) ______________________________

________________________________________

________________________________________

4. Has the information in this report led to any quantitative savings as far as man-hours/contract dollars saved, operating costs avoided, efficiencies achieved, etc.? If so, please elaborate.

________________________________________

________________________________________

5. General Comments (Indicate what you think should be changed to make this report and future reports of this type more responsive to your needs, more usable, improve readability, etc.) __________

________________________________________

________________________________________

________________________________________

6. If you would like to be contacted by the personnel who prepared this report to raise specific questions or discuss the topic, please fill in the following information.

Name: ______________________________

Telephone Number: ______________________________

Organization Address: ______________________________

______________________________

______________________________

— — — — — — — — — — — FOLD HERE — — — — — — — — — — —

Director
US Army Ballistic Research Laboratory
ATTN: DRDAR-BLA-S
Aberdeen Proving Ground, MD 21005

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

BUSINESS REPLY MAIL
FIRST CLASS PERMIT NO 12062 WASHINGTON,DC

POSTAGE WILL BE PAID BY DEPARTMENT OF THE ARMY

Director
US Army Ballistic Research Laboratory
ATTN: DRDAR-BLA-S
Aberdeen Proving Ground, MD 21005

— — — — — — — — — FOLD HERE — — — — — — — — — — —

END

DATE
FILMED

9 — 83

DTIC